EXPLORING ALGORITHMS TO SCORE CONTROL POINTS IN METROGAINE EVENTS

by

Wilhelmina Adriana van Hoepen

submitted in accordance with the requirements for the degree of

MASTER OF SCIENCE

in the subject

OPERATIONS RESEARCH

at the

University of South Africa

Supervisor: Professor EG Jones

Co-supervisor: Professor CJ Swanepoel

February 2018

Declaration

Name: Wilhelmina Adriana van Hoepen
Student number: 0464-830-7
Degree: MSc (Operations Research)

I declare that EXPLORING ALGORITHMS TO SCORE CONTROL POINTS IN METROGAINE EVENTS is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I have not previously submitted this work, or part of it, for examination at Unisa for another qualification or at any other higher education institution.

SIGNATURE: DATE: 23 February 2018

Acknowledgements

Not to us, LORD, not to us but to your name be the glory, because of your love and faithfulness.

Psalm 115:1

I acknowledge God Almighty, my heavenly Father, and Jesus Christ, my Saviour, and the Holy Spirit, my Helper, for without them, I can do nothing.

I would like to acknowledge the following people and thank them for believing in me, and for the role they played in the completion of this study:

- To my supervisors, Professor Jones and Professor Swanepoel, thank you for your invaluable time and input, and your endless patience.
- To my children, Nico, Rona, Salome and Jaco, for your unwavering support and encouragement.
- To my friend, Amelia, who was always only a phone call away.
- To my colleagues and COD at the Department of Decision Sciences, who supported me and allowed me the space to work on this research.
- To Rob Pratt of SAS, for your assistance with the software.
- To Lisa de Speville, who graciously shared her knowledge and granted me permission to use the Zoo Lake metrogaine map.

Abstract

Metrogaining is an urban outdoor navigational sport that uses a street map to which scored control points have been added. The objective is to collect maximum score points within a set time by visiting a subset of the scored control points. There is currently no metrogaining scoring standard, only guidelines on how to allocate scores. Accordingly, scoring approaches were explored to create new score sets by using scoring algorithms based on a simple relationship between the score of, and the number of visits to a control point.

A spread model, which was developed to evaluate the score sets, generated a range of routes by solving a range of orienteering problems, which belongs to the class of $\mathcal{NP}$-hard combinatorial optimisation problems. From these generated routes, the control point visit frequencies of each control point were determined. Using the visit frequencies, test statistics were subsequently adapted to test the goodness of scoring for each score set.

The findings indicate that the score-visits relationship is not a simple one, as the number of visits to a control point is not only dependent on its score, but also on the scores of the surrounding control points. As a result, the scoring algorithms explored were unable to cope with the complex scoring process uncovered.

Key terms: orienteering scoring problem, metrogaine events, orienteering problem, $\mathcal{NP}$-hard problem, scoring of control points, route choice, route planning, spread model, adjusting scoring approach, generating scoring approach, goodness of scoring, scoring metrics.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In this first chapter, the problem of scoring control points in metrogaine events is introduced and described, providing the rationale for this study. Metrogaine as a sport is first discussed to place the problem in context and provide the necessary background to understand the problem setting. The aim and objectives are given, together with the scope, in order to clearly delimit the study. A selection of the findings from the pertinent literature in the research problem area is also presented in this chapter. Because the research design and methodology followed in this study required a detailed description, these are addressed separately in Chapter 2. In conclusion, Chapters 2 to 6 are briefly outlined to provide the reader with a roadmap for this document.

1.2 Background

Metrogaining is a relatively new and unknown sport in South Africa. The first South African metrogaine event was held in April 2011 (AdventureLisa, 2011). Metrogaining belongs to a group of outdoor sports that requires navigation between control points, shown on a map, on foot. Orienteering, score orienteering and rogaining also fall into this group of navigational sports; they all require navigational skills using maps, but are set in different types of terrain. The word "metrogaining" is compounded from "metro", as in urban, and the last part of the word "ro-gaining".

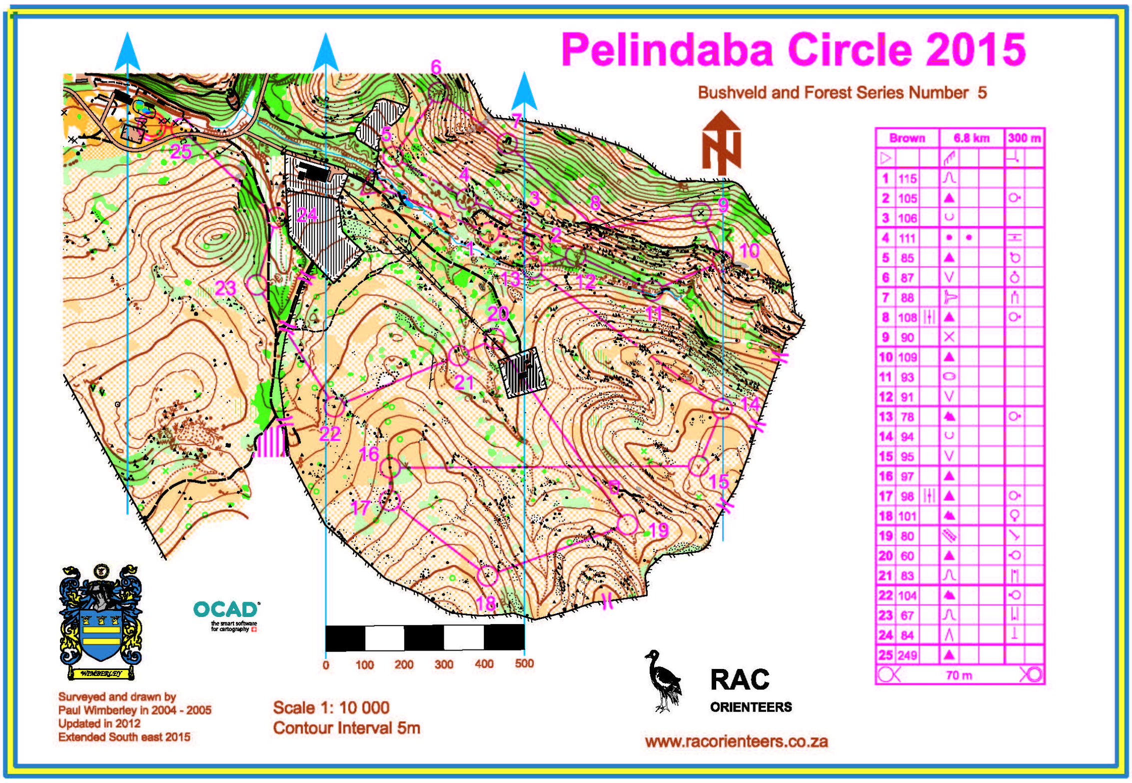


Figure 1.1: An orienteering map (Wimberley, 2015)

Metrogaining has developed out of rogaining, orienteering and score orienteering and incorporates elements from these sports. All of these sports combine a physical element of running/jogging/walking with mental elements of map-reading and decision-making throughout the event (Quantico Orienteering Club, 2014). A closer look at orienteering, score orienteering and rogaining will shed some light on the sport of metrogaining.

Orienteering as a sport developed in the Nordic countries in the 1930s and is the oldest sport in this group of navigational sports. The International Orienteering Federation (IOF) was founded in 1961 with 10 founding member countries. In 2017, the IOF had 71 member federations, with four member federations from Africa: Kenya, Mozambique, Somalia and South Africa. The four official orienteering disciplines are foot, mountain bike, ski and trail orienteering. Foot orienteering is the discipline relating to metrogaining and the one referred to as orienteering in this study.

Orienteering comprises a timed race where participants have to navigate between given control points, using only a map and a compass (International Orienteering

Federation, 2015). The sequence in which the control points must be visited is predetermined. The control points on the orienteering map are connected and numbered sequentially. The emphasis is on the navigation (route choice) between the control points. The competitor finishing in the shortest time, having visited all the control points in the correct order, is the winner (International Orienteering Federation, 2015). Participants compete as individuals at varying difficulty levels from beginner to elite. Starts are staggered with competitors setting off at constant time intervals of around two minutes. The event map is handed to the competitor at the start.

Orienteering takes place in diverse terrains such as plantations, open fields, nature reserves, and university and school grounds. An example of an orienteering map is shown in Figure 1.1 on page 2 (Wimberley, 2015). The numbered circles on the map indicate the position of the control points, which are connected by lines so as to be able to determine the location of the next control point quickly. However, the connecting lines do not indicate the route to be followed, as each competitor has to decide which route to take between the control points. The triangle on the map indicates the starting point and the double circle the finishing point. An orienteering map is a topographical map with a selection of prominent features, which assist with route choices by showing variations in the runnability and visibility of the terrain. This is done by using contour lines, and showing water features and different vegetation types, which are indicated by different colours. These colours are standardised by the IOF Map Commission (2017). For example, the darker the green on the map, the denser the vegetation with lower runnability. Dark green is effectively impassable. The colour white indicates forests or plantations with good runnability and visibility. The map also includes control descriptions, which describe the physical feature where the control point, marked by a white and orange orienteering kite, is placed (see Appendix A.1 on page 107 for a photograph of such a kite). The control point is positioned at the centre of the control circle on the map. The control description symbols are also standardised by the IOF International Orienteering Federation (2017). For example, a solid triangle is the symbol for a boulder and a solid circle represents a knoll.

Score orienteering is a variation of orienteering and can also take place on varying terrains. In this type of event, there are more control points than can possibly be visited within the allotted time and the competitor visits a subset of the control points, each with an associated score, within a predefined time limit (Orienteering USA, 2015). The control point scores are based on the distance of the control point

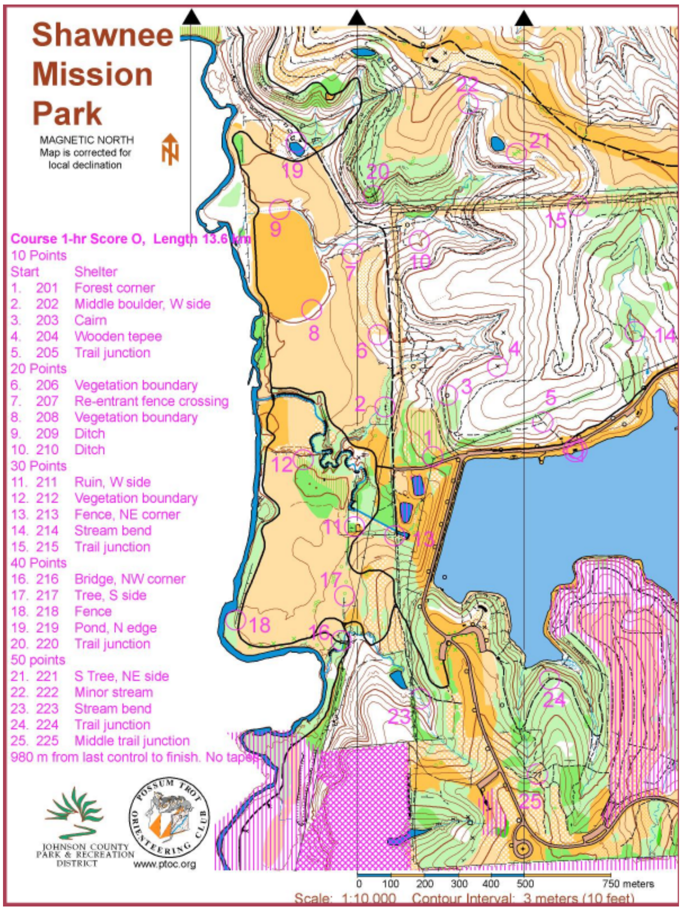


Figure 1.2: A score orienteering map (Possum Trot Orienteering Club, 2017)

from the start/finish and the technical difficulty involved in getting to the point (Velichko, 2004). Each control point may only be visited once and competitors have to select which control points to visit and in what order. Each competitor plans their own sequence of control points to visit to maximise their total score within the set time limit and their physical fitness and ability. Route choices must be made when navigating between the selected control points and a score penalty applies if the set time is exceeded. The competitor with the highest overall score is the winner. In score orienteering there is a mass start with all the competitors starting at the same time. As a result, owing to the set time limit, competitors all finish roughly at the same time.

An example of a score orienteering map is shown in Figure 1.2 (Possum Trot Orienteering Club, 2017). The map is similar to a normal orienteering map with contour lines, vegetation types and a selection of prominent features. The map scale is important as it assists competitors in selecting their sequence of control points within

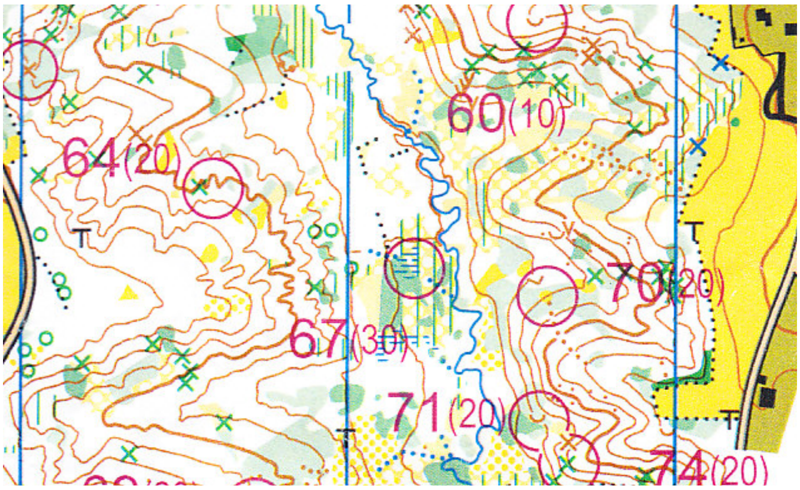


Figure 1.3: Score orienteering map with scores in brackets (Leibnitzer AC, 2017)

the distance they are able to cover in the allotted time. The numbered circles on the map, indicating the control points, are not connected as in orienteering, because each competitor determines their own sequence by selecting a subset of the control points. The starting point (triangle) and in score orienteering the finishing point (double circle) are also indicated on the map and are usually the same point.

In score orienteering, the control points are scored. There are different ways in which these scores can be shown on the map, some easier and quicker to grasp than others. One way is to list control points according to their score adjacent to the map, as shown in Figure 1.2. This makes it easy and quick to read the scores, but integrating the position of the control point with its score is difficult. Another way is to indicate the score adjacent to the control point number, as shown in Figure 1.3 (Leibnitzer AC, 2017) and Figure 1.4 (Bratt, 2015). Here it is easy to see the control point position with its associated score, but more map detail may be lost under the overprint. The scores could also be given indirectly through the control number, using a simple relationship between digits and the score (British Orienteering, 2016). An example of this method will be addressed in more detail below when metrogaining is discussed.

The set time limits for score orienteering are in the order of 60 to 120 minutes. Longer time versions (of up to 24 hours) of these types of events are called rogaining.

The sport of rogaining developed in Australia in the 1970s and is seen as the endurance version of score orienteering (International Rogaining Federation, 2012). It has gained prominence as an adventure racing sport in which competitors cover

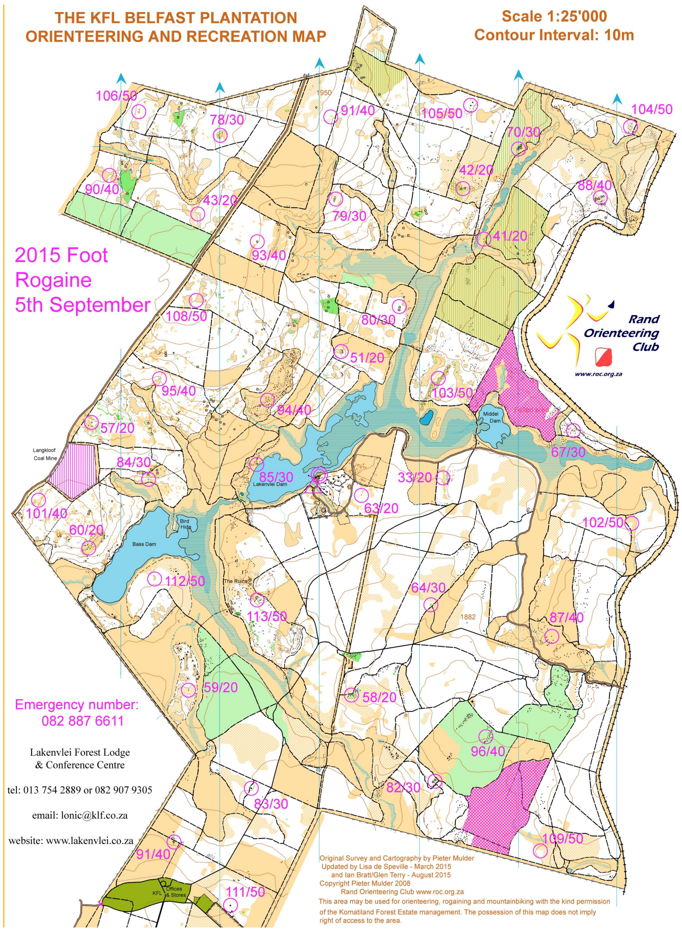


Figure 1.4: A rogaining map (Bratt, 2015)

long distances while navigating cross-country (International Rogaining Federation, 2015). It is a team sport with two to five members in a team. Rogaining is very similar to score orienteering, with the main difference being the length of the set time limit. For championships, the duration is 24 hours, but shorter versions down to three hours are common (International Rogaining Federation, 2015). In rogaining, the control points are called check points and may have different scores. As in score orienteering, rogaining has mass starts and there is no marked route. Each team has to select its own subset of check points and then navigate between them in the order selected by the team. Figure 1.4 shows an example of a rogaining map, which covers a much larger area than score orienteering, due to the much longer defined time period. Exceeding the time limit leads to disqualification.

Metrogaining is a specialised form of score orienteering, which takes place in an urban setting where the possible routes between the scored control points are confined to roads. These events are presented during daytime or at night. Competitors

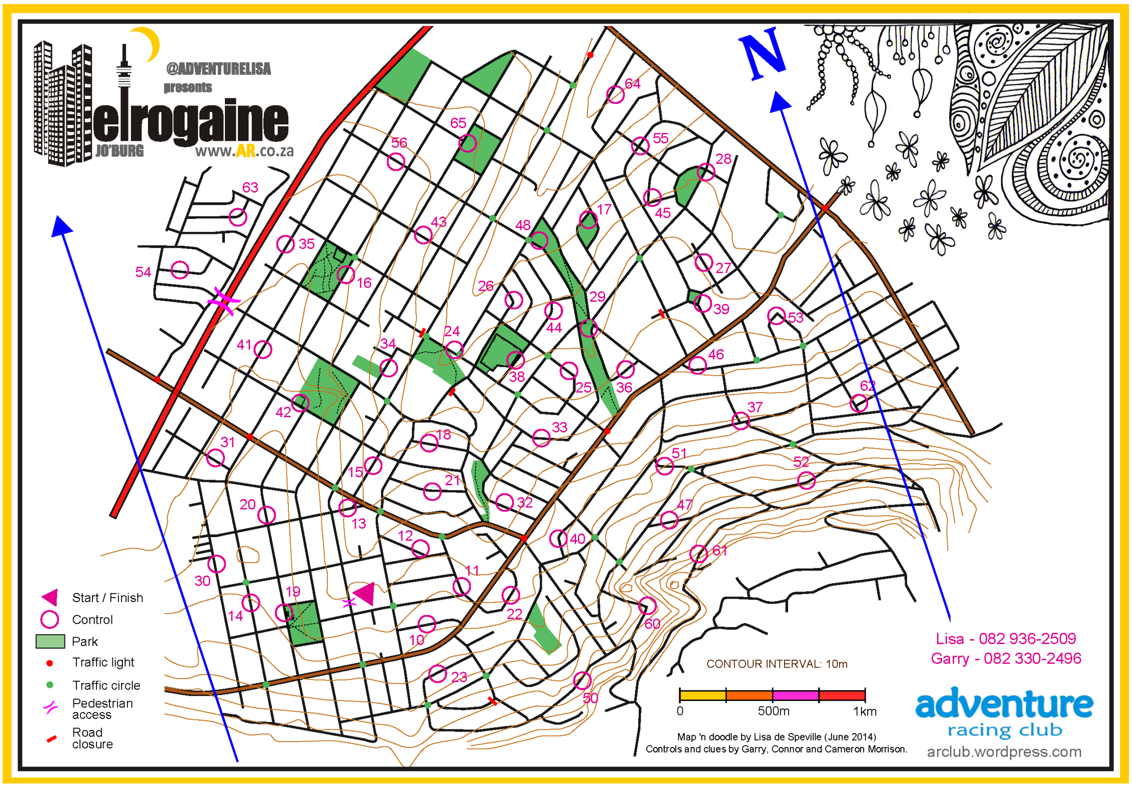


Figure 1.5: A metrogaining map (De Speville, 2015)

may enter as individuals, but in South Africa teams consisting of two members are preferred at night events. As in score orienteering, the aim is to collect the maximum number of points by visiting as many scored control points as possible only once within the set time period. As there are more control points than the fastest runners would be able to cover in the set time, teams are forced to select a subset of control points. Participants may choose between a time period of 60 minutes (short course) or 90 minutes (long course) to enter at a specific event. There is a mass start with the start and finish being at the same location and participants will finish either one hour or 90 minutes after the start. Participants are penalised by subtracting ten points for every minute they are late.

With the terrain confined to urban areas, a metrogaining map looks very different from the score orienteering maps shown in Figures 1.2 and 1.3. A metrogaining map is a street map with the identifying features removed. An example of a typical metrogaining map is shown in Figure 1.5 (De Speville, 2015). The scale is clearly indicated on the map to assist competitors in planning a route. All street names and significant features (like petrol stations) have been removed from the map, but traffic lights, traffic circles, road closures and pedestrian access are indicated.

Green areas are shown as well as contour lines, which assist in determining the steepness of a chosen route. The solid pink triangle indicates the start/finish. On the map, the position of a control point is indicated by a circle with its control point number adjacent to it. Unlike score orienteering events, at metrogaine events there are no physical markers indicating the control points. Instead, each team receives a separate clue sheet with questions corresponding to each control point number. Selecting the correct multiple-choice answer on the clue sheet is a team's proof of having visited the control point (AdventureLisa, 2011). An example of a two-page clue sheet can be viewed in Appendix A.2 on page 108.

In metrogaining, as in score orienteering, competitors walk, jog or run along a self-selected sequence of scored control points using a map. Control points have different point values (scores) assigned to them by the planner according to their distance from the start and technical difficulty. On the map, the scores of the control points are indicated by the left (first) digit of the control point number in multiples of tens, the so-called block values. For example, control point 38 has a score of 30 and control point 52 has a score of 50. This is a quick and easy method to obtain the score of each control point without taking up any additional space on the map, which could lead to overprint problems. However, with a two-digit control number, this method introduces a constraint in that it allows a maximum of just ten control points for each multiple-of-ten score. This constraint could influence the scoring of control points.

In Section 2.5, the routes and spread of teams in metrogaining are modelled; hence, it is considered useful here to describe what happens at such an event. At the start of a metrogaine event, each team receives a map and a clue sheet. First, the team has to do route planning before setting off to visit the selected sequence of control points. Keeping the aim of maximising its overall score and finishing in time in mind, a team estimates the total distance it could cover in the set time period. This distance will be different for each team and will depend on a team's speed (run, jog or walk) and fitness. Using the map scale and its distance estimate, the team starts planning a route. (Although no electronic devices may be used to assist in route planning, they may be used in tracking a team's route (De Speville, 2015).) There are no set rules for planning a route and teams may use different strategies, which are not addressed in any depth here. One strategy is to start by determining the furthest control point that could be visited by halving the estimated distance. (Usually, the further away from the start a control point is, the higher its score.) Several possibilities may be identified on the map in the process. For each possibility, the score of the furthest control point and its surrounding control points are examined. The team could identify an area with a concentration of the highest scores for the team's estimated distance. The furthest control point in this area would then become the team's turn-around point. The route to get to this turn-around point could then be planned, looking to pick up as many score points as possible by visiting interspersed control points along the way. Once a sequence has been decided on, the team sets off, making appropriate route choices and navigating between the control points as they go. As each control point in the selected sequence is visited, the correct answer on the clue sheet corresponding to the control point number is selected. It is essential for a team to check the remaining distance against the remaining time regularly to see if it is still on schedule. If not, the team needs to revise its route along the way in order to keep within the time limit. At the finish, the team's time is recorded, the clue sheet marked and the overall score of the team calculated.

With the practical background information in place, the literature is visited next for published information on topics related to the research problem.

1.3 Orienteering problem (OP) related literature

As described above, metrogaining is an urban form of the score orienteering event (SOE). In score orienteering, the aim is to select and visit as many scored control points as possible to maximise the total score within a given time limit, using a map indicating the control points, and travelling on foot. Each of the control points may only be visited once. Tsiligirides (1984) was the first to formulate a score orienteering event, which he called the generalised travelling salesman problem (GTSP). Golden et al. (1987) were the first to use the term "orienteering problem" for the score orienteering problem described by Tsiligirides. The term, *orienteering problem* (OP), is the one that is most widely used in the literature.

In the following subsection, the classical OP is discussed, as well as its variants, the different classes of OP solution methods, and other topics related to the OP that are directly applicable to this study.

1.3.1 The orienteering problem (OP)

The classical OP was first described in an article by Tsiligirides (1984), in which two approximate (heuristic) methods for solving the problem were presented. He described the SOE and noted some similarities between this type of event and the travelling salesman problem (TSP). This led to the formulation of the generalised travelling salesman problem, a variant of the TSP, describing a score orienteering event. Tsiligirides described the GTSP as a generalised SOE where the starting node and the finishing node are different.

According to Gutin and Punnen (2002), the TSP is probably the best known combinatorial optimisation problem. In the TSP, a travelling salesman has to find the shortest route (minimum distance) to visit each city in a given set of cities once. The travelling salesman starts at and ends in the same home city. The distances between the cities are known. In graph theory, the TSP can be defined as a graph G, where $G = (V, E)$ with V the vertices or cities, and E the edges connecting the cities. Each edge has an associated cost c_{ij}, the cost (or distance) of joining city i to city j in G. If c_{ij} is equal to c_{ji}, then the graph is called an undirected graph. If all the edges in a graph are bidirectional (c_{ij} is *not* equal to c_{ji}), then the graph is directed. The TSP is to find a Hamiltonian cycle (a cycle that visits each vertex of the graph once only) that would minimise the total cost of the cycle.

Unlike in the TSP, in an SOE not all the nodes have to be visited, only a selection of the nodes. If the travelling salesman knows the sales he could make at each city beforehand, he could plan his route to maximise the total sales. At the same time, he could limit the total route distance he travelled in a specified time period, by selecting a subset of the cities. Tsiligirides called this variant of the TSP the generalised travelling salesman problem (GTSP). To describe a score orienteering event as a GTSP in graph theory, the nodes are the control points and the edges the distances between the control points. The sales represent the control point scores. In terms of the GTSP, the objective is to select a subset of control points (which have to include the starting and finishing control points) in order to maximise the total score, subject to the total distance to be covered in the specified time limit. When Golden et al. (1987) presented his centre-of-gravity heuristic for the GTSP, he used the term "orienteering problem" (OP), which is favoured over that of GTSP in the literature.

Being a variant of the TSP, the orienteering problem is grouped with other variants

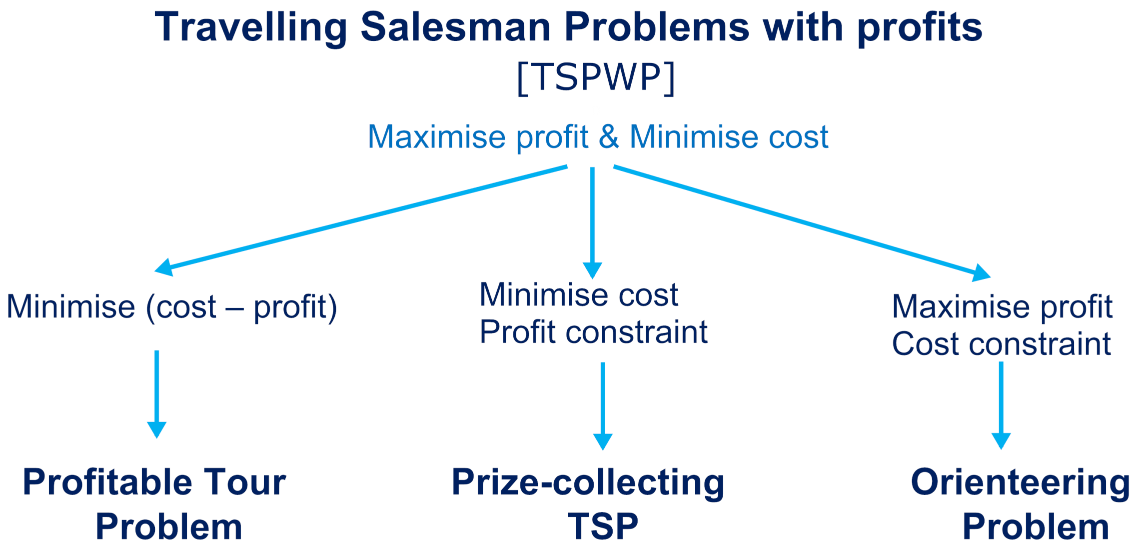


Figure 1.6: Generic subgroups of travelling salesmen problems with profits

known as the travelling salesmen problems with profits (TSPWP). This group is a variant of the TSP, having a very specific structure. Feillet et al. (2005) divide this TSPWP group into three generic subgroups according to the way the two objectives of total profit and travel costs are formulated. In Figure 1.6, these TSPWP subgroups are shown and it is clear why Gutin and Punnen (2002) saw the orienteering problem as the "dual" to the prize-collecting travelling salesman problem (PCTSP). In the OP the objective is to maximise the profit, subject to the cost being less than a preset value. In the PCTSP, the objective is to minimise the cost, subject to the profit being greater than a preset value.

In the knapsack problem, another combinatorial optimisation problem, a knapsack has to be filled with a given set of items. There are N different items available, each with its own weight and value. Items must be chosen from the given set to yield the greatest total value, subject to the limited weight capacity of the knapsack. The cost (distance) constraint in the OP is similar to the weight constraint in the knapsack problem and is also called the knapsack constraint by Feillet et al. (2005). Therefore, the OP could also be seen as a combination of the TSP and the knapsack problem, two classical combinatorial problems.

Other lesser known names for the orienteering problem are the selective travelling salesman problem (Laporte and Martello, 1990), (Gendreau et al., 1998b), the maximum collection problem (Kataoka and Morito, 1988) and the bank robber's problem (Awerbuch et al., 1998), (Arkin et al., 1998).

Ramesh et al. (1992) describes the orienteering tour problem (OTP) as an orienteering problem where the start and end points are identical, as in metrogaining. The OTP is also known as the rooted orienteering problem (Arkin et al., 1998). These terms are descriptive, but are seldom used in the literature.

Exact algorithms are described for the orienteering problem, but heuristics and metaheuristics are generally used when the number of nodes is large and solution times become too long.

NOTE: OP vs SOP terminology
As mentioned in Section 1.2, there are four official disciplines of orienteering as a sport (foot, mountain bike, ski and trail orienteering), where control points are *not* scored and the sequence in which they must be visited is given. In the literature, however, Golden et al. (1987) called the *score* orienteering event problem, where the control points are scored and the sequence in which they are visited must be chosen, the orienteering problem (OP). A more appropriate term to use for this problem is the score orienteering problem (SOP), as is done by Kataoka et al. (1998).
The reader should be aware of this possible confusion as the better-known term OP (instead of SOP) is used in this study.

1.3.2 Exact OP algorithms

The orienteering problem (OP) is known to belong to the class of $\mathcal{NP}$-hard combinatorial optimisation problems. Both Golden et al. (1987) and Laporte and Martello (1990) used simple reductions for proving that the OP belongs to the $\mathcal{NP}$-hard class. Angelelli et al. (2014) proved that the orienteering problem (without service times) can be solved in linear time $O(n)$ when the graph is a path or cycle. When service times are added, the OP cycle can be solved in $O(n^2 T_{\max})$ time and the OP path in $O(nT_{\max})$ time. Thus, the OP is, theoretically, solved exactly within a reasonable time, if the number of nodes is relatively small. In practice, exact methods are an option for solving orienteering problems. However, factors that should be taken into account are a researcher's view of what a reasonable time is, the complexity of the type of OP taken on, and the available computing power.

Lately, however, there has been a shift in the literature away from searching for more efficient exact methods for solving the OP, to research in heuristics and metaheuristics for OP and its variants. In the latest survey on OP by Gunawan et al. (2016), not a single new exact OP (not OP variants) approach was reported. According to the first OP survey by Vansteenwegen et al. (2011), the last research on new exact OP approaches was done in the late nineties of the previous century. This might be attributed to OP and its variants being applied to much more complex problems where a "good" solution is sufficient and an optimal solution a luxury, within the required response time.

There are two main approaches (with variations) recorded in the literature for exact OP solutions, namely, the branch-and-bound algorithm and the branch-and-cut algorithm, integrated in integer linear programming (ILP) or mixed-integer linear programming (MILP) formulation (Feillet et al., 2005).

The branch-and-bound algorithm enumerates points in a search space, defined by upper and lower bounds, which are recalculated in the enumeration process. The adapted branch-and-bound algorithms differ in the way the constraints are relaxed to compute the bounds. Gensch (1978) proposed a branch-and-bound algorithm relaxing the distance constraint in a Lagrangian way, solving up to 30 nodes for undirected graphs. Similarly, Kataoka and Morito (1988) also used Lagrangian relaxation, but took advantage of the special structure of the score function of the orienteering problem, solving up to 10 nodes for directed graphs. Kataoka et al. (1998) used the same Lagrangian relaxation, but added the constraint that a control point not visited can no longer have any descendent control point, to strengthen the relaxation. Laporte and Martello (1990) used a branch-and-bound approach that combined the special structure of the score function and the distance constraint of the orienteering problem, solving up to 70 nodes in undirected and 90 in directed graphs. The additive approach of Fischetti and Toth (1988) used different bounds sequentially after doing a Lagrangian relaxation of the distance constraint and solving the resultant assignment problem using a specialised algorithm. This approach resulted in solving up to 40 nodes in undirected and 100 nodes in directed graphs. Pekny and Miller (1992) used the same approach as Fischetti and Toth in relaxing the distance constraint using Lagrangian methods, but solving the resultant assignment problem using a different, specialised algorithm. The result was that up to 200 nodes were solved in directed graphs. Ramesh et al. (1992) also used the Lagrangian relaxation of the distance constraint to improve the quality of the bound by enforcing only one edge, thus leaving the first node (control point) to solve the orienteering problem with up to 150 nodes, both in undirected and directed graphs.

Gendreau et al. (1998a) developed an ILP using a branch-and-cut algorithm with valid inequalities for undirected graphs, solving up to 300 nodes. Also for undirected graphs, Fischetti et al. (1998) used a two-stage branch-and-cut approach, avoiding branching in the first stage by adding conditional cuts, thus resulting in a sparse graph. In the second stage, branching is then used on the sparse graph, solving up to 500 nodes for undirected graphs.

In all the exact solution procedures mentioned above, the following aspects should be kept in mind. The reported number of nodes solved for, was within a maximum CPU time limit set by the different researchers, varying from 100 CPU seconds (Laporte and Martello, 1990) to 18 000 CPU seconds (Fischetti et al., 1998). In most cases the computers used were specified. These computers were the ones that were available and used twenty years ago. The results would look different with today's computers if the same comparisons between the data sets, the CPU time and the number of nodes solved for were repeated.

1.3.3 OP heuristics and metaheuristics

When an optimal OP solution is not essential, a large number of nodes is involved, and time is of the essence, then heuristic procedures are used in solving the $\mathcal{NP}$-hard orienteering problem to obtain a sub-optimal solution. A heuristic is a procedure for solving a problem more quickly than an exact procedure, but produces only an approximate solution, without knowing if the approximate solution is a good solution or not. Heuristics provide a trade-off between optimality and speed, and are often routinely used in real-world applications.

Approximation algorithms are heuristics used to find approximate solutions for $\mathcal{NP}$-hard problems like the orienteering problem. A bound on the error of the sub-optimal solution for some approximation algorithms can be developed from theory, giving an idea of how good the solution is (Feillet et al., 2005). Different approximation algorithms have been developed and analysed for their complexity in special cases of the orienteering problem. Blum et al. (2007) and Bansal et al. (2004), for example, analysed the complexity of approximation algorithms for the OP where both the start and finish nodes are fixed, while Chekuri et al. (2012) analysed undirected OP graphs. Arkin et al. (1998) conducted a complexity analysis for the approximability of the OP where the start/finish is the same node and the route is a cycle.

Insertion, deletion, path extension, resequencing and substitution are all basic components that have been used alone or in combination, as classical heuristic solution procedures for orienteering problems. Cycling may be induced when combining some of these components (Feillet et al., 2005). The purpose of the heuristic procedures for the orienteering problem is to balance the collected score and the added distance for each additional control point included. Tsiligirides (1984) introduced two classical heuristic solutions to the orienteering problem, using sweep-based and path-extension procedures. Golden et al. (1987) developed a heuristic using insertion and resequencing, while John Mittenthal (1992) used insertion and deletion, and Laporte and Martello Laporte and Martello (1990), only insertion. Keller (1989) used path extension and resequencing, while Laporte and Martello (1990) used only path extension in another heuristic. Dell'Amico et al. (1998) use insertion, resequencing and substitution. Chao et al. (1996) used partitioning-based procedures as a heuristic.

Although heuristics can be very efficient, they do sometimes get trapped in local optima and cycling. To overcome these problems metaheuristics are used. Metaheuristics are generally regarded as problem-independent methods that can be applied to solve a wide range of problems, including the OP and its variants. In the literature, numerous metaheuristics procedures have been used to solve orienteering problems. A selected number of metaheuristics are mentioned here. The efficiency of the selected heuristic algorithms is not discussed here, only examples of their application of the OP and its variants are given. The techniques include: evolutive algorithms using learning and randomness (Golden et al., 1988); tabu search by Ramesh and Brown (1991), Gendreau et al. (1998b) and Fischetti et al. (1998); artificial neural networks by Wang et al. (1995); deterministic annealing by Chao et al. (1996); simulated annealing by Sylejmani et al. (2014); genetic algorithms by Tasgetiren and Smith (2000) and Ostrowski and Koszelew (2011); ant colony optimisation (ACO) by Liang et al. (2002), and ACO combined with neighbourhood search by Schilde et al. (2009).

Heuristics and metaheuristics for the OP and its variants are well researched. To choose an efficient metaheuristic for the OP or an OP variant is a challenging task, but in practice will be guided by the specific application and the available software.

1.3.4 Variants of the orienteering problem (OP)

When Tsiligirides (1984) first described the score orienteering event problem, he called it the generalised travelling salesman problem (GTSP), because of the similarities between the two problems. There has been a lot of interest and research in the variants and applications of the OP since then, as a number of challenging practical applications can be modelled using OP and its variants. One of the first surveys of applications modelled on the OP was compiled by Vansteenwegen et al. (2011). Later, an updated survey of OP variants and applications was published by Gunawan et al. (2016). These two papers give a comprehensive overview of OP-related research and provide a fascinating view on the development in this area, hence they should be scoured if information on definitions, solution approaches, benchmark instances and practical applications are required.

From the perspective of this research where the classical OP is used, detailed information on all the current OP and OP variants and their applications is not required. By listing some examples of the OP variants in broader practical application areas, the emphasis is placed on the classical OP as the origin of all these diverse applications.

- Orienteering problem (OP)
  Job scheduling (Gensch (1978) and Pekny et al. (1990)); inventory routing (Golden et al., 1987); control theory (Ramesh and Brown, 1991); single-ring networks in telecommunication (Thomadsen and Stidsen, 2003); tourist trip design (Souffriau et al., 2008)

- Team orienteering problem (TOP)
  Recruitment (Butt and Cavalier, 1994); team orienteering (Chao et al., 1996)

- Orienteering problem with time windows (OPTW) and team orienteering problem with time windows (TOPTW)
  Inventory routing and tourist trips with specific opening hours for deliveries and visits (Vansteenwegen et al., 2011)

- Generalised orienteering problem (GOP) (multiple scores for different attributes)
  Tourist trips with different factors or categories of attractions (Schilde et al., 2009), (Wang et al., 2008), (Geem et al., 2005)

- Orienteering problem with stochastic profits (OPSP)

Obsolete inventory in auto industry (Ilhan et al., 2008)

- Capacitated team orienteering problem (CTOP)
  Transportation services (Feillet et al., 2009)

- Time dependent orienteering problem (TDOP)
  Public transport; tourism (Garcia et al., 2013)

The list above contains no OP variant used in sport, except for the team orienteering problem (TOP), where team members split up and each member selects his or her own subset of control points with no overlapping control points between the team members, claiming a specific control point score only once. All team members must still finish within the time limit. In score orienteering events and metrogaining, while participants also compete in teams, the team members have to stay together all the time. Although it would be possible to use such a team set-up as described by Chao et al. (1996) in a score orienteering event, this has not been done to the author's knowledge. TOP is a sport application in theory, but not in practice.

1.3.5 Route choice and route planning

Route choice is a core part of all orienteering disciplines, using only a map and a compass (International Orienteering Federation, 2015). In the official forms of orienteering, route choice involves choosing which route to follow, while navigating the terrain between the control points in the given sequence. To choose the best possible route, competitors must read and interpret the map, which shows the topological characteristics and the vegetation of the terrain. The key aspects, which separate the winners from the runners-up in this sport, are to be able to consistently choose the best possible route, navigating precisely while moving at speed between control points. The trade-off between concentrating on the terrain and reading the map while running determines the quality of the decision-making and navigation (Eccles, 2008).

Modelling route choice in orienteering is very difficult as there can be as many route choices as there are competitors. Hayes and Norman (1984) used dynamic programming to generate route choices and site controls to assist the planners of orienteering events. It was, however, impractical to try and evaluate all possible route choices and several assumptions were made. These included transforming and simplifying the map to a grid of 250 metre by 250 metre blocks on the ground, rules to generate travel times, and dynamic programming to find the shortest paths. The authors found that the simple model was able to assist planners with siting controls and planning orienteering routes. The ground work for preparing the attributes of the grid was time-consuming, however, and might deter planners from using the model on new maps.

In score orienteering, rogaining and metrogaining events, however, competitors not only have to deal with route choice, but also with route planning; that is, selecting a subset from the set of available control points and selecting the order in which the control points will be visited (the route). Once the route is planned, deciding on the route choice and navigating between the control points follow. Subsequently, not only do competitors try to optimise these route aspects without any electronic devices, but they also have to reconsider them along the route so as to keep within the set time limit.

Gordon (2006) has looked into route planning strategies for rogaining, a long distance, cross-country version of score orienteering events, as discussed in Section 1.2. He investigated effective route planning strategies for competitors to use during a rogaine event, using an enumeration algorithm. Some of the assumptions Gordon made to model the rogaining problem included that teams stay on paths between control points, they finish within the time limit, and they travel at a constant speed regardless of the terrain or direction of travel. The result was a set of robust strategies that was easy for competitors to apply and worked well, if the assumptions were reasonable for the specific event.

The effect of uncertain speed in rogaining, compared to the constant speed assumed in Gordon's model, was investigated by Dye et al. (2010). They used a stochastic, branch-and-bound enumerative approach, introducing a set of speed scenarios, described by a speed and its corresponding probability. The speed could be an average speed, and distances could be adjusted according to the terrain. Similar results for the deterministic and stochastic versions were obtained, but the sequence of the control points became very important in the stochastic version. In terms of solution times, the number of control points had a far bigger effect than the implementation of uncertain speed. According to the authors, their stochastic optimisation algorithm would seem to be viable, but needs to be improved and tested more widely.

Modelling route choice and route planning in orienteering and score orienteering events remains a challenge and is mostly dealt with by simplifying the choices considerably through assumptions. In metrogaine events, the route choices between the control points are assumed to be along roads only. In practice, however, to minimise the time moving between control points, competitors look for the shortest path between the control points. In the next section, shortest path algorithms are addressed.

1.3.6 Shortest paths

The shortest path problem is one that is often encountered in network models, and the score orienteering event model is no exception (Cherkassky et al., 1996). Once the route has been planned for a specific score orienteering event, the route choices between the selected control points must be made. In a metrogaine event, the route choices are limited to roads, which results in a limited and finite number of possible paths between the control points. Competitors want to follow the shortest road paths between the selected control points to optimise their chosen route.

The shortest path problem is a relatively simple problem, but has a high complexity and in general runs in $O(n^3)$ steps (Johnson, 1973). The problem has been studied extensively in the literature due to its frequent application in combinatorial problems (Gallo and Pallottino, 1986), where algorithms are developed to improve the complexity of specific cases. The more popular classification of these algorithms is the label-setting and label-correcting methods, but Gallo and Pallottino (1986) also proposed a classification based on the structure rather than the behaviour of the algorithm.

Dijkstra (1959) published his well-known solution to the shortest path problem as a note on graphs. Dijkstra's original algorithm (a label-setting method) is only applicable to positive edge lengths and runs in $O(n^2)$ steps (Johnson, 1973). When Dijkstra's algorithm is modified to accommodate negative edge lengths as well, it becomes a label-correcting method (Gallo and Pallottino, 1986). Many variations of the label selection method of Dijkstra's algorithm are used, as described and evaluated by Gallo and Pallottino (1986) and Cherkassky et al. (1996). These evaluations were done on randomly generated networks.

The evaluation of 15 shortest path algorithms by Zhan and Noon (1998) on real road networks is of special interest for this study, which deals with shortest road paths between control points. Zhan and Noon neatly summarises the algorithms, their complexity and references in a table, should the reader require more specific information.

Both the studies by Cherkassky et al. (1996) and Zhan and Noon (1998) conclude that for nonnegative edge lengths, one of the Dijkstra variations should be considered.

1.3.7 The orienteering scoring problem

In the classical orienteering problem (OP) described in Section 1.3.1, the control points scores are given and fixed for a specific instance. In most applications and variants of the OP, it will merely be stated what these scores usually represent, a reward or profit which is collected upon visiting the node. For example, Butt and Cavalier (1994) described an application of a recruiter visiting schools to recruit athletes. In that case, the score would be based on recruiting potential. Another example, from the same authors, is a technician servicing customers, with the score taking customer importance and task urgency into consideration. In the tourist application from Vansteenwegen and Van Oudheusden (2007), scores represent tourists' preferences and attraction values in a variant of the OP. The detail of exactly how these scores are determined is seldom, if at all, mentioned. In this research the classical OP and how to determine the scores of the control points are studied, in what could be called the orienteering scoring problem.

In his original problem, Tsiligirides (1984), who originally defined what is now known as the OP, looked at how the scores could be reallocated as a "related problem". His approach was to have consistent standards across different types of orienteering events. In orienteering, the aim is to visit the given sequence of control points in the correct order in the shortest time. According to Tsiligirides, gold, silver and bronze orienteering time standards were set as a factor of the winner's time. For example, a competitor achieved a gold time standard if his or her time was less than 1,25 times the winner's time. In score orienteering events, Tsiligirides wanted to extend these standards to scores to promote consistent standards across different orienteering types. Accordingly, Tsiligirides looked at the relationship between the total score and the distance covered in an event. He reasoned that the total score achieved by a competitor should increase linearly and proportionally to the distance covered. To achieve consistent standards, Tsiligirides proposed that the intercept of the fitted linear regression line (between the total score and the distance covered) should fall within an interval of plus or minus 0,05 of the winner's total score. The intercepts of the fitted linear regression line for two score orienteering problems were subsequently tested by Tsiligirides and found not to be within the required interval. He attempted to see if he could change the intercept of the fitted linear regression line to fall within the required range by modifying the scores of the control points. He proposed a scoring algorithm wherein he combined the distance to a control point's nearest neighbour, the average distance from the specific control point to all the other control points, and some operational factors to limit the range of new scores. The scoring algorithm he proposed was not successful in achieving his goal, however. Although he suggested that other scoring algorithms might be more appropriate, he did not name or suggest any.

In the process of evaluating the performance of a branch-and-cut OP algorithm, Fischetti et al. (1998) proposed use of TSP benchmark data sets. As TSP nodes are not scored nodes, the authors described three different ways of scoring these nodes to transform these TSP benchmark data sets into OP data sets, where the nodes (control points) are scored. The three ways include: set all control point scores to the same value, randomise the scores in an interval from 1 to 100 (including both values), and assign bigger scores to the nodes further from the starting point, by using a ratio of the specific control point distance from the start, to the control point furthest away from the start.

Apart from these two instances of orienteering scoring algorithms, no others could be found in the literature. The lack of research in this area may be attributed to the fact that scoring is very problem specific and can seldom be generalised.

1.4 Description of the problem

In the sport of metrogaining, the control points on the metrogaine map are scored manually by the event planner. There are no formal standards, only informal guidelines for scoring control points, resulting in subjective scoring by each event planner. This makes the comparison of different metrogaine event results questionable. In general, the factors involved in scoring a control point are the distance from the starting point and the navigation difficulty in reaching the control point. Scoring algorithms are explored to establish whether the scoring process can be done consistently and quantitatively, using an objective basis. The significance of this research is in laying a foundation for a future scoring standard in the sport of metrogaining. The exploration of scoring algorithms will make a positive contribution to the sport of metrogaining and to operations research in general. Moreover, the exploration of algorithms addressing this $\mathcal{NP}$-complete problem is an interesting practical application of optimisation techniques in the field of sport. Were a standard to be set it would allow for the comparison of different metrogaine events, which could contribute to building and growing the sport.

1.5 Research scope

As discussed in Section 1.2, there are four official disciplines of orienteering (foot, mountain bike, ski and trail orienteering), and other specialised (unofficial) formats within these disciplines. In studying metrogaine events, a very specialised form of orienteering, the research scope is limited. The characteristics of a metrogaine event (in no particular order) are listed below:

- Metrogaining falls within the official discipline of foot orienteering, which means that a competitor has to walk, jog or run.
- A time limit is set within which competitors visit as many control points as possible to maximise their total score.
- Control points can be visited in any chosen sequence.
- Metrogaine events take place exclusively in urban areas. Controls points are accessed by road.
- The starting point is also the finishing point.
- In metrogaine events the control points are scored in multiples of 10, with the minimum score being 10 and the maximum 90.
- The unique way of identifying a control point and its score on metrogaine maps results in only a maximum number of *ten* control points having the same score. (See Section 1.2)
- A penalty of 10 points is deducted from the total score for every minute a team exceeds the time limit.

– Some time may be spent at a control point selecting the correct option on the clue sheet.

The time penalty is to force teams to keep to the time limit. In the control point scoring process, it is assumed that all teams will stay within the time limit and thus penalties are not included in this study. Another assumption is the one of no service time at the control point, that is, no time is spent at the control point selecting and marking an option, as it is possible to do so on the move.

There are two limitations that need to be looked at in more depth. The first is that the control points are all connected by roads, which simplifies the model as there is only one shortest route to reach a specific control point, and the distance is known or can be determined. Although this constraint is mostly true in an urban setting, there may be green areas within a suburban setting where control points could be visited by not following a road. Crossing such green areas could theoretically result in a large number of routes. If there is a minimal number of these non-road routes, the effect on the scoring of the control points may be minimal, but this does not form not part of this study and has not been tested. The second limitation that needs attention is the maximum number of ten control points having the same score on a specific map, which can be referred to as the block values constraint. This constraint makes the control point scores easily recognisable, but it might affect the scoring of the control points. Relaxing this constraint is not addressed in this study.

There are two aspects that should clearly be noted as not forming part of this study. The lack of a scoring standard for metrogaine events is a gap that was identified and it initiated this research. Therefore, there is no intention of generalising the research for application to other disciplines or forms of orienteering events other than metrogaining. The research is about *scoring* control points in metrogaine events. In this study the positions of the control points are given and fixed. Thus, the study is not about *positioning* control points or changing their positions.

1.6 Aim and objectives of the study

The aim of the study is to explore quantitative methods for scoring control points in metrogaine events in order to form a basis for setting a scoring standard in the sport. Scoring control points should be done in such a way as to encourage competitors to make different route choice decisions and spread them across the entire map area. Currently, the control points are scored manually, using one basic quantitative measure, combined with nondefined, individualised qualitative measures from each event planner.

The following research objectives are pursued in this study:

- To extract map data from the Zoo Lake metrogaine map as a case study
- To extract and manipulate additional road data in the Zoo Lake area
- To calculate different types of distances (using the above data) as scoring algorithm input to create control point score sets
- To model the spread and develop scoring metrics for metrogaine events
- To evaluate scoring algorithms for metrogaine events

1.7 Document outline

Following the description of the aim and objectives above, the outline of the chapters given below reflects the logical course of the research in order to achieve these objectives.

Chapter 2: Methodology
In this chapter the different scoring approaches are introduced. The scoring of the control points is described and the need for benchmark score sets is discussed. Team routes and spread are modelled, thus developing scoring metrics to evaluate the scoring algorithms.

Chapter 3: Data
The focus of this chapter is on the input data needed for this study. Maps are mathematically formulated as graphs, with vertices and edges, to assist with data descriptions and calculations. Map data and the different types of distances used in the scoring algorithms are discussed and calculated. The data accuracy is discussed and some theoretical background is included where needed. The chapter concludes with a summary of the methods selected for the data calculations, the software and the calculation accuracy used.

Chapter 4: Scoring control points
This chapter starts off with a look at the existing scoring method. Next, the orien-

teering problem (OP), which forms an integral part of the scoring algorithms and their evaluation, is formulated and discussed. The details of each of the adjusting and generating algorithms are then discussed and applied to create the different score sets. The chapter concludes with information on the benchmark score sets.

Chapter 5: Results
The visit frequency results of the different scoring algorithms and their resultant score sets are determined and presented graphically. In addition, the goodness of scoring (as a scoring metric) for the different score sets is calculated, analysed and discussed.

Chapter 6: Conclusion
The final chapter summarises the findings and conclusions on the different scoring algorithms that were explored. The contributions of this research are presented. Included in the suggestions for further research is the formulation of the orienteering scoring problem.

1.8 Conclusion

In this first chapter an overview of the relevant literature on the background to the specific problem setting was given. The problem, scope and objectives of this research were described. Concluding the chapter was the document outline, which summarised the chapters to come. The next chapter looks at the methodology of this study in detail.

Chapter 2

Methodology

2.1 Introduction

This research is about scoring algorithms and scoring approaches for metrogaine events. In this study, a scoring approach describes the method used to obtain control point scores. A scoring algorithm consists of specific steps during which an approach method is applied to determine the score of a control point. The control point score is an important factor in the route planning decisions made by participating teams, which want to maximise their total score. A team's route is defined by the control points selected and the order in which the team plans to visit these. Combining all the teams' selected control point visits shows how the teams are spread across the entire metrogaine map area during the event. By looking at different scoring algorithms, the intention is to see if control points can be scored quantitatively and objectively in a manner that is good enough to be considered a scoring standard for metrogaine event planners.

In this chapter, the research design and methodology are discussed in order to develop and evaluate different scoring algorithms. The data is discussed in depth in Chapter 3, therefore only a brief summary of the data is given in this chapter. Thereafter, two scoring approaches and the different scoring algorithms within each approach are described. This is followed by the modelling of the effect of control point scores on team routes, as well as team spread, in order to develop a scoring metric. The two goodness of scoring statistics that have been developed to determine the goodness of scoring, are discussed. The chapter concludes by considering the ethical aspects of this study.

2.2 Research design and methodology

This study is a non-empirical analysis of existing numerical data, relating to a physical sports event. The research design chosen for the analysis is a combination of case studies, secondary data analysis and model-building (Mouton, 2001). The control point data is the data source and is used to create scoring algorithms, which are analysed for their goodness of scoring. Previous work relating to scoring schemes by Tsiligirides (1984) and Fischetti et al. (1998), can be used as scoring algorithms. In order to develop a scoring metric to evaluate the scoring algorithms, model-building is required. Consequently, to formulate a workable model in this study, processes have been simplified while taking care to still reflect reality. The goodness of scoring has to be measurable to evaluate the scoring algorithms. As this is only an exploratory study, one example of a metrogaine map has been chosen as a case study to work with.

The methodology followed in this study is set out in the sections below and is summarised by the outline in Figure 2.1.

2.3 Data

In this exploratory study, the Zoo Lake metrogaine map was chosen as the example of a metrogaine event to focus on in this exploratory study. Lisa de Speville, the planner of the Zoo Lake metrogaine event, provided the map both electronically and in hard copy, and granted permission to use it (De Speville, 2015). This specific event was chosen as a case study as additional useful information was available from the planner.

As the primary data, the attributes of the control points on the map (latitude, longitude and elevation) were extracted, using 1map, a South African online geographical information system (1map, 2017). Using the attribute data, the shortest road distances between all the control points were calculated as input data for the scoring approaches and the spread model, using an online, free geographical map, OpenStreetMap (OpenStreetMap contributors, 2015) and SAS software. Two other

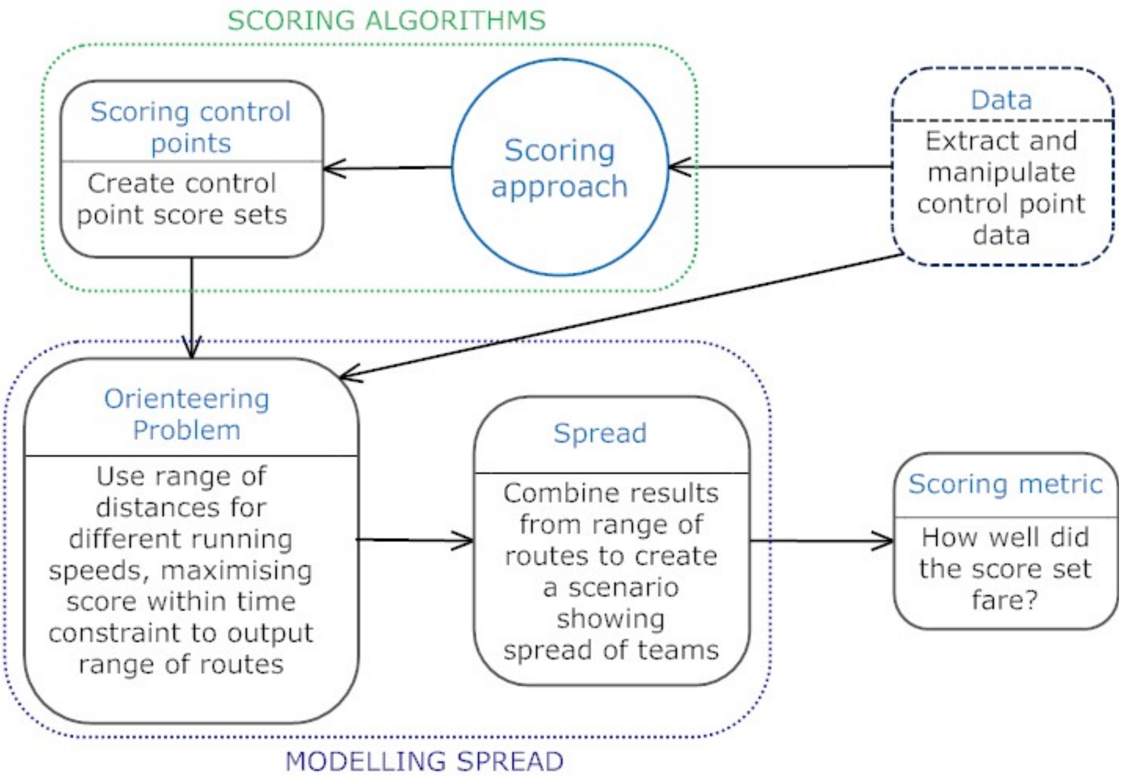


Figure 2.1: Methodology outline

types of distances, the geodetic and elevation adjusted road distances, were calculated as input to the generating scoring approach. The geodetic distance and the elevation adjusted road distance are defined in Section 3.4 on page 48 and Section 3.5 on page 49, respectively. Chapter 3 is dedicated to the extraction, preparation, formatting and calculation of the input data.

2.4 Scoring control points

In scoring the control points, two main approaches were explored, the adjusting scoring approach and the generating scoring approach. Using these two approaches, scoring algorithms were developed to create 24 new score sets.

2.4.1 Adjusting scoring approach

In the adjusting scoring approach, an initial set of scores, where all the scores are the same, was used as the reference score set. Adjustments were made to the scores based on the combined number of visits to the control points using the score set, `SAME`, which has the same score for all the control points. Three adjusted scoring algorithms were developed resulting in three new score sets, `SAME_I`, `SAME_II` and `SAME_III`. These scoring algorithms are described in Section 4.4 on page 62.

2.4.2 Generating scoring approach

Four different scoring algorithms were developed in the generating scoring approach. Each of these scoring algorithms has been summarised below:

1. A score for each control point was generated by using the three distance measures of global and local work getting to the control point. These distance measures are road distances, elevation adjusted road distances and geodetic distances, defined in Sections 3.3, 3.4 and 3.5, respectively. A set of five global to local work ratios (global work:local work) were used to calculate the total work in terms of each of the three types of distances. Fifteen score sets were subsequently created using these work scoring algorithms. Three examples of the score sets generated are `Road80G20L`, `Alt_adj50G50L` and `Geo20G80L`. A full description of these work scoring algorithms is contained in Section 4.5.1 on page 67.

2. The travelling salesman problem (TSP) score set, based on the travelling salesman problem, was generated making use of the shortest cycle obtained when visiting all the control points once (a Hamiltonian cycle). The points closest to the start (five points on both sides) were scored the lowest. The scores of the following ten control points were increased on both sides of the cycle (five points on each sides), until the last five were scored the highest. This TSP scoring algorithm is described in Section 4.5.2 on page 70.

3. The third generating scoring algorithm used, was the one proposed by Tsiligirides (1984). Although his reason for changing the scores differed from the ones in this study, his algorithm could be used. He generated a control point score by making use of the distance to its nearest neighbour (NN_d), and the average distance from the specific control point to all the other control points (Avg_d). To generate the score, he combined these two distances using different ratios and scale factors, this was then approximated to the nearest integer of five. The NN_d:Avg_d ratios used by Tsiligirides were 0,0:1,0;

1,0:0,5; 0,5:0,5 and 0,5:1,0. Although Tsiligirides found that the algorithm performed unsatisfactorily for his purpose, it was nevertheless decided to include an adaptation of Tsiligirides' algorithm in this study for the sake of completeness, as he was the first to propose a scoring algorithm in the literature. For this study, three NN_d:Avg_d ratios were selected to create three score sets, `TS0Si100Sii`, `TS50Si50Sii` and `TS100Si0Sii`. The Tsiligirides scoring algorithm is described in Section 4.5.3 on page 70.

4. Fischetti et al. (1998) generated scores in three different ways for instances in a TSP library in order to test the performance of their branch-and-cut, OP-solving algorithm. Although the aims for generating the scores in their study differed from those in this study, all the scoring algorithms could be used. In the first score generated by the authors, all the scores were set the same. In this study this has already been done in Section 2.4.1 and the score set is called `SAME`. The second score was generated in a random way. In this study, a chosen benchmark score set, `RANDOM`, has been similarly created, and is described in Section 4.6 on page 72. The third and last score generated by Fischetti et al. used a ratio of two distances, where both distances were measured from the starting control point, but with different ending control points. The one distance is measured from the start to the specific control point, and the other distance from the start to the control point furthest away from the start. For a specific map with its given, fixed control point positions, the latter distance would be a constant value. The details of this algorithm are described in Section 4.5.4 on page 71.

2.4.3 Benchmark score sets

Two data score sets, `RANDOM` and `EXISTING`, were selected as benchmarks sets. In the `RANDOM` set, random numbers were used to score the control points. Section 4.6 on page 72 describes the way the random numbers were obtained and the scores generated. A comparison of algorithmic calculated scores and random scores would indicate whether scoring algorithms add value to the process of scoring control points. A good scoring algorithm should score control points better than randomly selected scores. The other benchmark set, `EXISTING`, was generated by the planner of the Zoo Lake metrogaine event. Accordingly, a comparison of the planner's scores and algorithmic calculated scores would indicate whether the scoring algorithms in this study do better or worse than the current, manual scoring process used by metrogaine planners.

2.5 Modelling routes and spread

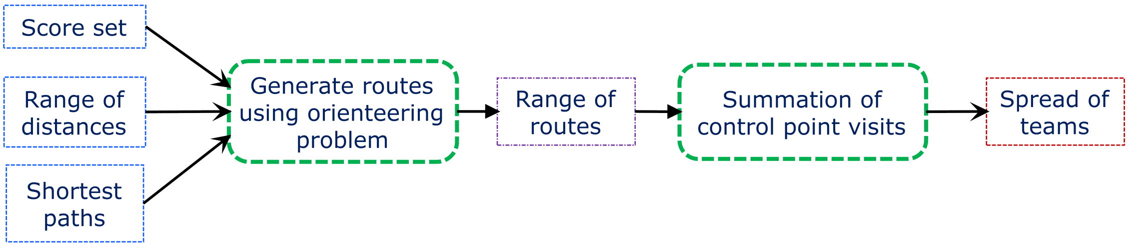


Figure 2.2: Modelling of spread

The competitors in metrogaine events include teams of walkers, joggers, runners and serious runners. In a metrogaine event, each team has a different average speed, depending on how fast they move on foot. As the event time is set, distance rather than speed is used in the modelling of team routes. To model these team routes, a range of distances was chosen to generate a range of team routes by applying the orienteering problem (OP), using one specific distance at a time. The output from each of these orienteering problems is an optimal team route of visited control points, which would result in the maximum number of score points collected for the specific distance within the set time. For each distance in the range of distances, the control points visited would be known. By summing all the individual control point visits over the range of routes for each control point, modelled the spread of the teams across the metrogaine map was modelled. This team spread model is represented diagrammatically in Figure 2.2 and formulated in conjunction with developing scoring algorithms. The value of building such an extended model lies in explaining the process of scoring the control points and the effect different scores have on team spread.

Of the three input streams shown on the extreme left in Figure 2.2, two are calculated and discussed in other chapters. The shortest paths are discussed and calculated in Section 3.3.1 on page 46 in the chapter on data. The score sets used in this model are created in Chapter 4, starting from Section 4.4 on page 62, onwards. The third input stream, the range of route distances, as well as the rest of the model elements shown in Figure 2.2, are discussed below.

Select a range of distances to model the different average speeds of teams

To model the different average team speeds, a range of distances is needed as discussed above. The specific average speeds of teams at an event are unknown, but this is not needed in this study. The range of distances only has to model the range of realistic speeds that can be expected at any metrogaine event.

For older adult walkers, covering 5 km in an hour is considered a brisk walking speed for exercise, and therefore 5 km is chosen as the shortest distance in the range of distances (Parise et al., 2004).

When planning a metrogaine event, to ensure that a subset of control points is selected, the number of control points should be more than the fastest team could visit in the specified time limit. The travelling salesman problem (TSP) distance is the shortest route visiting each control point once, starting from and returning to the start. Using SAS software, the Zoo Lake TSP distance was determined as 28,2 km. This distance should be longer than the fastest team would be able to cover in 90 minutes. To check that this would indeed be the case, information on the South African athletics record for the 25 km road race was obtained. The South African record holder for the 25 km road race ran at a speed of just under 20 km per hour (Wenig, 2010). This record speed, run on roads similar to those in metrogaine events, translates into around 20 km for the 60 minutes time limit and 25 km for 90 minutes. The maximum distance covered by the winning team in the Zoo Lake metrogaine event was 21,1 km in the allotted 90 minutes (De Speville, 2015). Therefore, this TSP distance can be taken as the maximum distance in the range of distances, as no team will be able to cover this distance in the longer specified time limit of 90 minutes.

Starting from the minimum value of 5 km, distances were incremented by one kilometre, up to 20 km. The range was completed by adding 21,1 km (the winners' distance), 25 km (South African record distance), as well as the maximum distance of 28,2 km. These 19 distances were used in the model to represent the different average team speeds.

Generate a range of team routes

The orienteering problem (OP) is a combination of control point selection and determining the shortest path between the selected control points. The result of solving the OP with the required input (see Figure 2.2) is an optimal route, that is, the control points to be visited and the order in which the control points should be visited, staying within a given distance and collecting the maximum total score for the distance. Therefore, by running the OP with each of the distances in the range of distances, a range of optimal routes will result. If this procedure is repeated for all the different score sets, a range of optimal routes for each of these score sets will be generated. These ranges of optimal routes could then be used to determine the spread of the teams for the Zoo Lake map for each score set. In turn, the team spread could be used to evaluate the score sets.

At first glance, however, assuming the routes taken by the teams will be optimal and using these optimal routes to evaluate a specific score set, may appear unrealistic at first sight. In reality, teams aspire to select optimal routes for their average speeds. The closer they get to an optimal route, the closer they get to the maximum total score they could achieve within the teams' limitations. This is the aim of this sport, and the mind set with which teams participate. Therefore, the use of optimal routes when evaluating score sets is founded on a practical and sound base.

Although the OP is a $\mathcal{NP}$-hard problem (Golden et al., 1987), the number of control points at metrogaine events is less than 100. This relatively low number of points makes it possible to solve the orienteering problem exactly. Finding software to solve the orienteering problem was the next step. It was not necessary to search for the software that would most efficiently solve the orienteering problems, however, as any software that was available and could solve the problems within a reasonable time, would have been adequate. Not only did SAS software satisfy both criteria, but the author was already skilled in using SAS. Therefore, SAS was chosen to solve the orienteering problems. SAS uses a row generation approach (relaxing connectivity) to solve multiple mixed integer linear programming problems in order to solve orienteering problems exactly (Pratt, 2015b).

For each of the 26 score sets, a range of 19 routes was generated, using SAS to solve the 494 orienteering problems.

Combining the visited control points over the range of distances
For each score set, the range of distances was used as input to run the orienteering problems in order to obtain the range of optimal routes. From these routes, the control points visited were recorded and summed over the range of routes, per control point. The result was the combined number of visits to each control point over the 19 distances. The number of visits not only showed how many times a control point was selected when using a specific score set, but also showed how spread out the teams were for the score set used.

The spread of the teams across the map area was determined for each of the score sets created and the two benchmark score sets. The range of possible combined number of control point visits is from 1 to 19. All the control points are visited once in the TSP distance, resulting in a minimum value of 1. If a control point is visited by all the teams in every one of the 19 distances, the maximum value of the combined number of visits is 19.

Using the combined visits to each control point, scoring metrics were developed to evaluate the score sets and their associated scoring algorithms.

2.6 Scoring metrics

By changing the scores of the control points, the number of visits to each control point also changes. For instance, if a specific control point was not selected to be visited by a team, and the score of that control point were increased, the team would be tempted to visit the control point to increase their total score. If the team decides it is worth their while to visit this control point with its increased score, the number of visits to this control point would increase, and vice versa. Thus a scoring metric using the combined number of visits to each control point would determine how good a particular score set fared in terms of the spread.

Ideally, in this model, the number of visits to each control point should be the same. This would ensure that the teams are spread out evenly across the whole map area. To find a test statistic to indicate the goodness of scoring, two dispersion statistics were considered.

2.6.1 Coefficient of variation (CV)

The average number of visits is used to set up the scoring metric to see how well a specific score set fared. The deviation of the observed number of visits from the average number of visit per control point gives an indication of how well the score set and its scoring algorithm performs.

In descriptive statistics, measures of variation (or dispersion) are used to describe the variation in a data set (Bradley, 2007). Commonly used absolute measures of variation are the range, the quartile deviation (or interquartile range), the variance and the standard deviation. Although both the variance and the standard deviation take all the values of the data set into account when measuring the variation from the average of the data set, the standard deviation is preferred because it has the same units as the average.

If the variation across different data sets is compared using the standard deviation, the averages should be close. If the averages differ, relative measures of variation are more appropriate for a reliable comparison of variation. The most common relative measure of deviation is the coefficient of variation (CV), which is defined as

$$\mathrm{CV} = \frac{s}{Average\ frequency}.$$

The standard deviation, s, is defined as

$$s = \sqrt{\frac{\sum(Observed\ frequency - Average\ frequency)^2}{n-1}}$$

where n is the number of control points, and frequency is the number of visits per control point.

The ideal performance of a scoring algorithm occurs when the observed control point visits are as close as possible to the average frequency of control point visits. Therefore, the lower the CV value, the better the performance of the scoring algorithm.

2.6.2 Distribution uniformity (DU)

Distribution uniformity (DU) is an indicator used in evaluating the performance of irrigation systems for crops in order to improve crop quality and yield. This indicator measures how uniformly the irrigation sprinkler system distributes water over a crop area. The DU is defined as the ratio of the average water applied to the driest quarter of the crop area, and the average water applied to the whole crop area (Burt et al., 1997). Although overwatering the crop area is wasteful, underwatering is the critical aspect focused on in the DU.

In practice, the uniformity of water distribution can be measured by arranging same size tins in a regular grid pattern across the crop area (Growcom, 2013). The depth of the water in each numbered tin is measured after irrigation and the tins are then ordered according to the water levels from shallow to deep. The average of the lowest 25% readings is calculated and then divided by the average depth of the water levels in all the tins to yield the DU. If the irrigation water is applied 100% evenly, the DU is one.

In evaluating the score sets, a variation of DU was applied. Ideally, the visits to each control point visit should be the same when evaluating the performance of a scoring set. This is similar to the ideal objective of having the same water depth in each tin when evaluating the performance of irrigation systems. The number of visits to the control points was ordered from the lowest to the highest. The average of the lowest 25% control point visits, divided by the average of all the control point visits was calculated as the lower end distribution uniformity, $\mathrm{DU_{LV}}$. In this case, both the low numbers and the high numbers of control point visits of interest. To obtain the ideal situation where all the control points have the same number of visits, low number of visits should become higher *and* the high number of visits should become lower.

Therefore, the lower end distribution uniformity for control point visits is defined as

$$\mathrm{DU_{LV}} = \frac{\text{Average of the lowest 25\% control point visits}}{\text{Average visits over all the control points}},$$

and the upper end distribution uniformity for control point visits is defined as

$$\mathrm{DU_{UV}} = \frac{\text{Average of the highest 25\% control point visits}}{\text{Average visits over all the control points}}.$$

As both the lowest and highest 25% control point visits, and by how much they differ from the ideal, are important, the value $(\mathrm{DU_{UV}} - \mathrm{DU_{LV}})$ gives an indication of how far the score set is from the ideal score set. The lower the $(\mathrm{DU_{UV}} - \mathrm{DU_{LV}})$

value, the better the performance of the scoring set is.

The $(\mathrm{DU}_{\mathrm{UV}} - \mathrm{DU}_{\mathrm{LV}})$ value does not use all the available data set values, only 50% of the values. If the middle 50% of the data values are close to the average, the $(\mathrm{DU}_{\mathrm{UV}} - \mathrm{DU}_{\mathrm{LV}})$ value should compare well to a statistic using all the data values.

The goodness of scoring was determined for all the score sets using the two statistics, the coefficient of variation and the adapted distribution uniformity.

2.7 Ethical considerations

This research project which involves the exploration of scoring algorithms, lies within the field of applied mathematics and the subfield of operations research. It is a quantitative study and does not involve human participants, animals, or other living or genetically modified organisms. According to Unisa's ethics guidelines, this study presents negligible ethical risk.

Ethical clearance for this research project was granted by the Research Ethics Review Committee of the School of Economic Sciences at Unisa on 27 May 2014. A copy of the Ethics Approval document (Ref#:2014_CEMS_SES_002) is attached in Appendix B on page 111.

2.8 Conclusion

This chapter on methodology discussed the way in which the research was planned and would be conducted. In the process, the spread of teams was modelled. The test statistics for quantitatively evaluating score sets were also selected and adapted.

The next three chapters are central in exploring scoring algorithms for metrogaine events. This exploration starts with Chapter 3 which examines the data used in this study.

Chapter 3

Data

In the previous chapter, the research methodology and design of this study were described. The methodology was outlined and summarised diagrammatically in Figure 2.1 on page 29. From the diagram it can be seen that everything starts with the data. The aim of this chapter is to define, introduce and describe the map data, which is the first input step of the study. The focus of this chapter is on aspects relating to the data source, the Zoo Lake metrogaine map. Some theoretical aspects relating to the different types of distance data will also be touched on. Furthermore, the accuracy of the data used in this study will be discussed and determined. By the end of the chapter, both the map data and the distance data should be well described and should be presented the format required for use as input data.

The primary data available from the metrogaine map is data on the control point locations, the existing control point scores, and the distances between the control points. The data contained in the Zoo Lake metrogaine map appears in a map format, which is not a computational format. To be able to analyse the map data, it has to be extracted and converted into a numerical format. Each control point has a unique geographical location, described by its latitude and longitude coordinates, as well as its elevation. The associated control point scores can be read from the map and will be dealt with in Chapter 4, the chapter on scoring. The road distances between the control points are obtainable from the map. Another two distances, geodetic and elevation adjusted road distances, can also be obtained from the map indirectly. These different types of distances are described and dealt with in the sections below.

Several software packages were used to digitise, manipulate and format the data as required. These software packages are briefly mentioned here, however the role and use of each will be elaborated on at the point of application. OpenStreetMap (OSM) (OpenStreetMap contributors, 2015) and 1map (1map, 2017) were used for the geographical map data, as well as the road distances, while uMap (uMap, 2015) was used as the visual interface for OSM. Both OSM and uMap are freely available online and were chosen for that reason. The online geographical information system (GIS), 1map, was developed in South Africa and is available free of charge for academic purposes. In addition, SAS software (SAS, 2012) was used in calculating the geodetic distances and solving the orienteering problems in Chapters 4 and 5. SAS was chosen because the author was already familiar with the software, an academic student licence was available, and the required program codes for the orienteering problems could be obtained. MS Excel was used for the spreadsheets, which were used extensively, together with the other software packages to prepare input data sets, collect output data sets and perform low level calculations.

3.1 Map data

The Zoo Lake metrogaine map, shown in Figure 3.1 on page 41, was the primary data source for this study. The map was made available to the author in hard copy and electronically in PDF format by the event planner, who also gave permission for its use (De Speville, 2015). A full page image of the map is available on page 54.

Mathematically, the relationships between the control points on the map can be described in graph theory (Biukaghazadeh, 2013). The map is then called a graph which can be described using concepts from graph theory: vertices (the control points) and edges (the connecting road distances). By assigning a positive weight (a distance, which is positive) to the edges, the graph becomes a positive weighted graph. The Zoo Lake graph is also an undirected graph, that is, if the distance from control point i to control point j, is the same as the distance from control point j to control point i. Formally, the graph is formulated as G, with $G = (V,E)$ where $V = \{v_1, \ldots, v_n\}$, the set of n vertices and E the set of undirected edges. In the case of the Zoo Lake graph, the number of vertices (n) is 56, including the start/finish vertex. Also, all of the vertices are connected by an edge to all other vertices. As the edges are undirected, we only have one unique edge connecting all of the vertices to each other, with no duplicates. Therefore, the number of edges is:

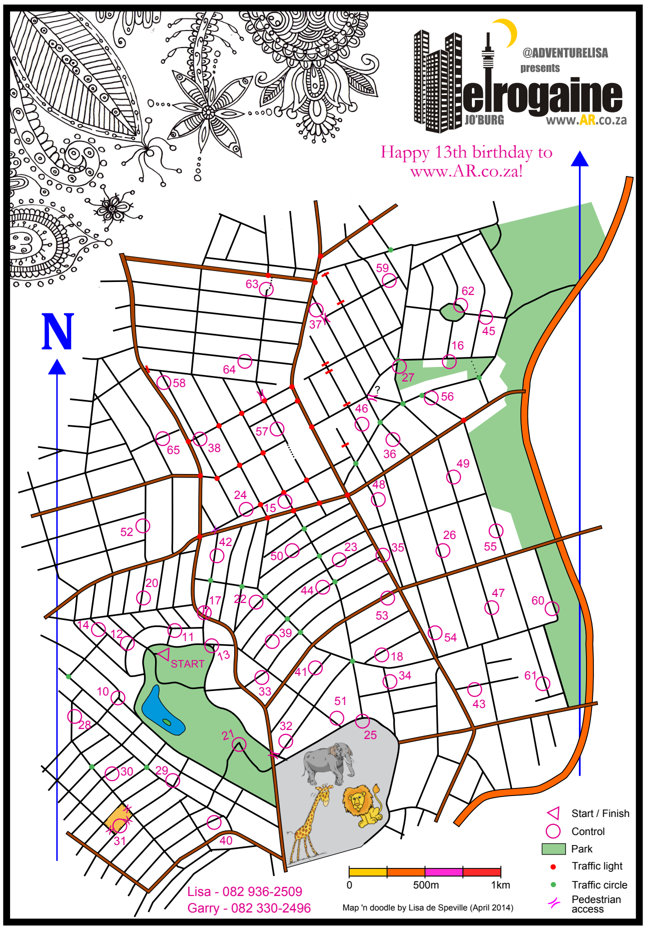


Figure 3.1: The Zoo Lake metrogaine map (De Speville, 2015)

$(56 \times 55) \div 2 = 1\,540$. All that remains to describe the graph uniquely is to determine the 1 540 elements in the set of weighted edges. As previously mentioned, three types of distances (edges) have been used in this study. Therefore, the elements of the three sets of weighted edges need to be determined. The shortest road distances between all the control points are dealt with in Section 3.3, the elevation adjusted road distances in Section 3.4 and the geodetic distances in Section 3.5.

Describing and formulating the map as an undirected, positive weighted graph, assisted in selecting an appropriate algorithm to solve the shortest path problem in Section 3.3. The features of an undirected, positive weighted graph were also used in the SAS OP coding.

The positions of the control points, which are addressed in the next section, are required to determine the shortest road, the elevation adjusted road and the geodetic distances.

3.2 Control point attributes

The geographical position of the control points on the metrogaine map in Figure 3.1, are indicated by small, pink circles, with their unique, two-digit numeric identifiers (in pink) adjacent to the circles. The pink triangle shows the start/finish. The exact position of the control point is located at the centre of the circle. These control point positions, which were required later on in the study, had to be digitised in a suitable format for software input.

To obtain the control point attributes, 1map was used. The Zoo Lake metrogaine map area was matched and selected in 1map by the maximum and minimum latitude and longitude values for the area. (The maximum latitude selected was $-26{,}167°$and the minimum $-26{,}153°$, with the negative value indicating a latitude south of the equator. The maximum longitude selected was 28,046°and the minimum 28,012°.) The control points, including the start/finish, were then manually placed and marked on a layer of the demarcated map in 1map, using the positions as marked on the metrogaine map. In 1map, all the control points were moved from their exact positions to positions in the middle of the adjacent road, perpendicular to their original positions. This was done to simplify the road distance calculations, while keeping the model realistic.

NOTE ON TERMINOLOGY
Elevation is the vertical height of the local ground surface of the earth above sea level.
Altitude is the vertical height an object above the local ground surface of the earth.
In this study, the correct term is *elevation*, as the ground level height is of interest. However, the automated output from some software uses *altitude* for this height. Whenever the term *altitude* occurs in this study, *elevation* is implied.

One of the 1map features is the ability to export spatial data from layers. The spatial data on the control point layer included the latitude, longitude and altitude of each marked control point. These attributes were exported as a CSV file, which could be opened in any text editor and MS Excel. An SAS data set of the control point attributes was created by importing the attributes from MS Excel. This SAS data set formed part of the input to determine the three different distances used in this study. The list of the Zoo Lake control point attributes obtained from 1map and used throughout the study, is available in Appendix C on page 113. Please note that there is no Control Point 19 (CP19) on the Zoo Lake metrogaine map.

3.2.1 Accuracy of control point attributes

Latitude and longitude

The latitude and longitude values (in degrees) exported from 1map were given to thirteen decimal places. To obtain some understanding of the accuracy of latitude and longitude values, let us assume that the earth is a sphere. The earth's radius at the equator is 6 378,137 km (Moritz, 2000). The circumference of the earth at the equator can be calculated as 40 075,017 km. Therefore, the distance covered by one degree of latitude at the equator is the circumference divided by 360 degrees, which is about 111 km. At a latitude of 26 degrees south (which corresponds to the Zoo Lake area), the earth's circumference is different from the circumference at the equator, and the distance of one degree latitude changes as a function of the circumference. By using trigonometry and geometry, the non-equatorial circumference is calculated as the equatorial circumference multiplied by the cosine of the latitude (Fenton, 2001). At 26 degrees latitude, the circumference is 36 779,032 km and the distance of one degree latitude is about 102 km. Therefore, from a theoretical view, using five decimal places for the latitude (and longitude) coordinates would result in the location of the control points being accurate to about one metre.

Some other information and practical issues on the accuracy and assumptions made regarding the distances and distance calculations of the Zoo Lake map have been considered, and are listed below.

- To get an idea of the range of distances involved, the road distance between the two closest control points (CP25 and CP51) on the map is 202 m and the shortest cycle distance selected for the orienteering problem is 5 km.
- In metrogaining, it is not necessary for competitors to go right up to the control point as is the case of orienteering events. As discussed in Section 1.2, at a metrogaine event teams prove that they have visited a specific control point by answering a question relating to its immediate surroundings. This can be done from varying distances, which vary between teams, as well as control points.

– In addition, as discussed in Section 3.2, the control points were placed in the middle of the roads, although teams usually do not (and should not) run in the middle of the roads due to traffic.

– The transfer of the control point positions from the Zoo Lake map to 1map was done manually. The planner also originally placed the control point positions on the Zoo Lake map by hand.

– A competitor taking one step in any direction at any time would move about one metre in distance.

Looking realistically at the spread model (see Figure 2.2 on page 32), and taking into account the effect and influence of all these aspects (individually and combined), it was decided that working with a distance accuracy in the order of one metre in this study would be sufficient. This meant that latitude and longitude values accurate to five decimal places were used in the distance calculations in this study.

Altitude

The altitude values (in metres) exported from 1map were also given to thirteen decimal places. These altitude values were obtained by 1map via the Google Elevation Service (Grobbelaar, 2015). Google Elevation Service snaps to the nearest point for which they have a known altitude. According to Wang et al. (2017), Google has not yet released any data on the accuracy of their elevation data. In a study done by Wang et al. on USA elevation data for transportation applications, they found the accuracy, using the mean absolute error, to be 1,32 m. For this study, the author selected a number of points on contour lines, reading off the elevation values from the contour lines on 1map. Next, the elevation values for these points on the contour lines were exported from 1map, using the Google snap values. In addition, the elevation of these points on contour lines were also manually read from Google Earth. These three elevation values for each of the 29 selected points on contour lines were then compared. The comparison in Figure 3.2 shows that although the absolute elevation differed, there is an almost constant difference at each point between the three elevation values. In this study elevation differences were used and not the absolute values. Therefore, using the Google snap values for the elevation differences in this study was deemed to be acceptable.

The elevation values obtained for the control points were on roads, and the difference

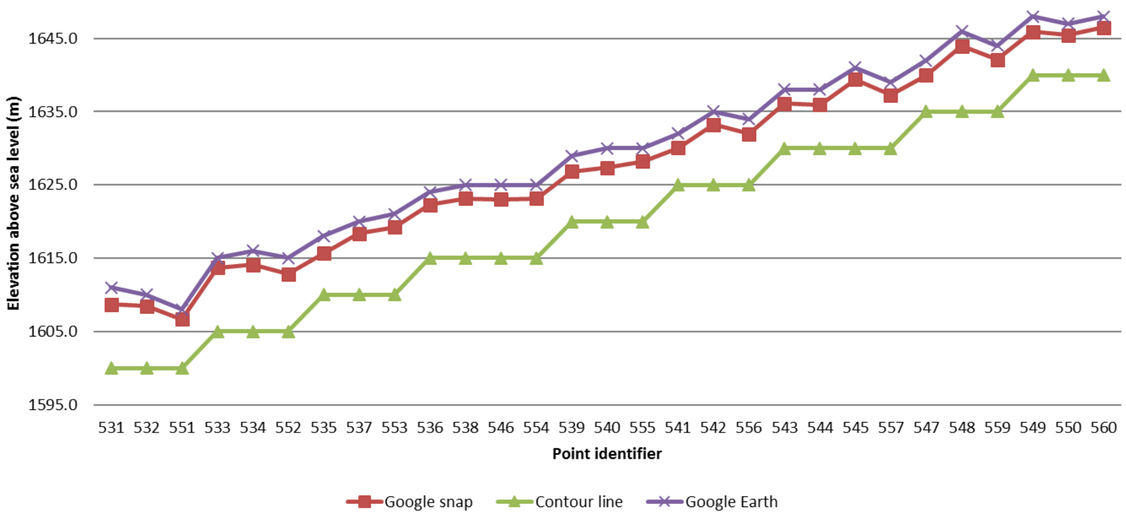


Figure 3.2: Comparison of elevation values

in elevation values has therefore been referenced to the road surface. However, competitors could go up a pavement or down into a pothole while moving between control points. Therefore, it was decided to use elevation values (in metres) to one decimal place in the elevation adjusted distance calculations.

3.3 Road distances

Owing to the urban nature of metrogaine events, the routes taken by teams have to be run on roads. Therefore, information on all roads connecting the control points on the map had to be collected. The information required included distances, intersections, traffic circles, and direction changes to describe the roads. Using these features enabled the actual road distance covered when moving between the control points to be determined.

The road information was sourced from OpenStreetMap (OSM). OSM is a project supported by the OpenStreetMap Foundation that uses crowdscourced, geospatial map data, which is freely available under an open licence. It was thus possible to obtain raw geographical map data for the Zoo Lake area. The data was accessed in OSM and exported by using the OSM export function, specifying the minimum and maximum values of the latitude and longitude of the specific Zoo Lake area. The exported data included all the data required to draw everything that can be seen on the OSM map. For this study, much of the exported data was superfluous; just the relevant road information had to be extracted, cleaned and formatted to be used as input to calculate the shortest road distances between all the control points. The process of extracting, cleaning and formatting the data is described in Appendix D, starting on page 115.

3.3.1 Shortest road distances

The road distance output from OSM described in Appendix D, is the required road distance information for the 147 roads on the Zoo Lake metrogaine map. This information was used to determine the shortest road distances between all the control points.

The shortest path problem is a well-known problem in graph theory (Biukaghazadeh, 2013). In general, the aim of this problem is to find how to move from a starting vertex, via other available interconnecting vertices, to an ending vertex, in such way as to minimise the sum of the distances between the starting and ending vertices. In this study, the starting and ending vertices are control points. The interconnecting vertices which identify road intersections, circles or other points where the road changes direction, such as bends or kinks, are called road vertices in this study. This is to distinguish them from control point vertices. In some cases a control point vertex could also be a road vertex, but that is usually the exception. In this study, the aim is to find the sequence of vertices that minimises the sum of the distances along the selected road vertices, starting and ending with two control point vertices; that is, to determine the shortest road routes and minimum road distances between any two control points.

Dijkstra's algorithm (1959) is a well-known algorithm for solving the shortest path problem. Although there are several different algorithms available for the shortest path problem, the algorithm by Dijkstra is quite famous and is still widely used (Biukaghazadeh, 2013). The original algorithm is to be used on positive weighted graphs only, but according to Biukaghazadeh, modifying the original algorithm to accommodate negative weighted graphs as well, is quite simple.

As the Zoo Lake graph is a positive weighted graph (see Section 3.1), there is no problem in using Dijkstra's original algorithm or any of its variants. SAS software was used to determine all these shortest road distances between any two control points. SAS uses a variation of Dijkstra's algorithm (SAS, 2015) to calculate these distances. The SAS code used can be found in Appendix J.1 on page 147. The SAS input used was the `ZL_map_data` file generated from OSM as described in Appendix D.1. The SAS input file contained all the required information of all the road and control point vertices of the Zoo Lake graph. The file contained 4 599 road vertices and 56 control point vertices, with 35 226 edges connecting the relevant vertices. The SAS output distance file, `ZL_arcs`, contained the 1 540 entries of the shortest road distances (edges) between all the control point vertices, as well as the sequences of the road vertices for each of these edges. Ninety-four of the 1 540 shortest road distances were selected to visually verify the results in uMap. With uMap, which is open source software, custom maps can be created using the created OSM layers of road and control point vertices, and road edges. More information on the uMap visualisation is available in Appendix D.2 on page 120. A uMap example of one of the shortest road distances between two control point vertices, together with the intermediate road vertices, is shown in Figure 3.3.

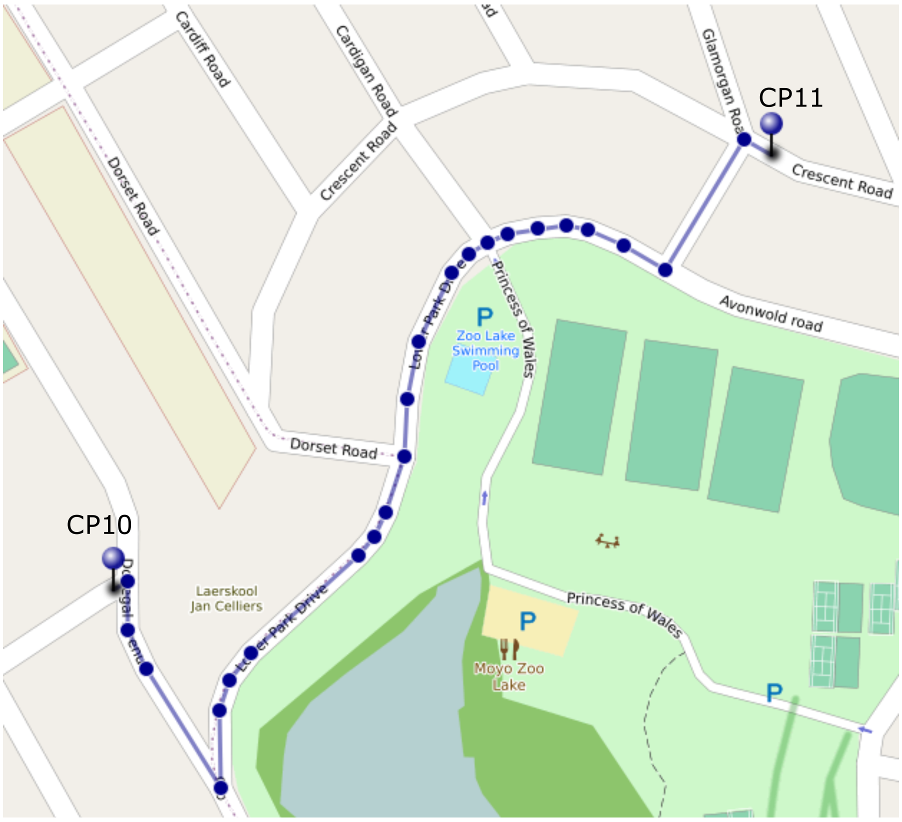


Figure 3.3: Path between CP10 and CP11 (Source: uMap, 2015)

One of the inputs to the orienteering problem (OP), as shown in Figure 2.2 on page 32, is the shortest paths. Therefore, these 1 540 shortest road distances from

`ZL_arcs` are used as input in all the orienteering problems, and also in some of the scoring algorithms in Chapter 4. From `ZL_arcs`, the values of 54 road edges between two control points were selected and listed in Table E on page 123.

Hereafter in this study, road distances refer to the *shortest* road distances between any two control points.

3.3.2 Accuracy of road distances

The accuracy of the road distances between the control points depends on the accuracy of the latitude and longitude coordinates of the road and control point vertices, which was dealt with in Section 3.2.1. The outcome was that in this study, distances accurate to the nearest metre would be worked with.

3.4 Elevation adjusted road distances

Three types of distances are used in the generating scoring approach in Chapter 4. One of these is the road distance which has been discussed in Section 3.3 above. Another type of distance is the geodetic distance which is to be discussed in Section 3.5. In this section, the second type of distance, the elevation adjusted road distance, is addressed.

Hikers and runners competing in mountainous navigation sports have to identify potential routes and then select the fastest route. These route choices often involve making a decision between taking a shorter but steeper route versus a longer but flatter route. For years these competitors have used Naismith's rule (Scarf, 2007) to estimate and compare route times for their potential choices. Naismith's rule gives an equivalence between ascent, and horizontal distance travelled. Scarf (2007), who has done a thorough study of Naismith's rule, recommends that males use the rule of 1 m ascent equivalent to 8 m of horizontal distance, and females an equivalence ratio of 1:10.

Although metrogaining takes place in urban areas on roads, the same equivalence principle of Naismith's rule can be applied. Timewise, it will take teams longer to cover a distance if the road is at an incline than if the road is flat. For simplicity sake, the equivalence ratio of 1:10 was used throughout this study. Road distances adjusted by using Naismith's rule of 1 m ascent equivalent to 10 m of horizontal distance are called elevation adjusted road distances, or in short, elevation adjusted distances. Naismith's rule makes no adjustments for descent, therefore descending road distances were not adjusted. The resulting issue of directed edges (distances) has been addressed in Section 4.5.1 on page 67, where these elevation adjusted road distances have been used in generating scores by measures of work.

In practical terms, this means that the elevation difference (in metres) between two specific control points was determined, and multiplied by ten. This equivalent horizontal distance was then added to the road distance (between the same two control points), as calculated in Section 3.3.1, to arrive at the elevation adjusted road distance between the two specific control points. These elevation adjusted calculations were done in MS Excel for all control points, where applicable. A subset of 54 elevation adjusted road distances (edges) is listed in Table E.1 on page 123 adjacent to the road distances.

3.4.1 Accuracy of adjusted road distances

According to Section 3.2.1, the chosen accuracy of the elevation difference is 10 cm. Using Naismith's rule, the equivalent horizontal distances were calculated and added to the road distances, which are accurate to one metre. Therefore, the accuracy of the elevation adjusted road distances would also be one metre.

3.5 Geodetic distances

Geodetic distance is the third and last type of distance used in the generating scoring approach in section 4.5.1. Geodetic distance is the shortest distance between two points identified by latitude and longitude coordinates, also called the geodesic distance (Karney, 2013). Some theoretical aspects relating to the calculation of geodetic distances are discussed below. In concluding this section, the geodetic distance formula chosen for this study is discussed.

The geodetic distance takes the curvature of the earth into account. Therefore, an earth model must be chosen. Assuming that the earth is a sphere, the geodetic distance may be calculated using the great-circle distance (using the law of cosines), or the haversine formula. Using the ellipsoidal earth model, the Vincenty formulae or the algorithms developed by Charles Karney could be used. These four formulae are discussed below.

3.5.1 The spherical earth model: the law of cosines

The great-circle distance formula using the law of cosines can be derived from spherical trigonometry as

$$d = R \cdot \cos^{-1}(\sin\phi_1 \cdot \sin\phi_2 + \cos\phi_1 \cdot \cos\phi_2 \cdot \cos(\lambda_1 - \lambda_2))$$

where

d is the distance between two points on the surface of the earth
R is the equatorial radius of the earth
$(\phi_1; \lambda_1)$ is the latitude and longitude of point 1, in radians
$(\phi_2; \lambda_2)$ is the latitude and longitude of point 2, in radians

According to the Geodetic Reference System (Moritz, 2000), the mean radius of the earth is 6 378,137 km. Substituting R with 6 378,137 km, the great-circle distance, d, would also be in kilometres.

Geodetic distances in the order of one metre or less should not be calculated using the great-circle formula. This is because the inverse cosine for small angles requires more significant digits than the 15 decimal precision digits of 64-bit floating point numbers computers can offer for meaningful accuracy at these small distances (Chamberlain, 2008), (Huber, 2011).

3.5.2 The spherical earth model: the haversine formula

The haversine formula originates from using an archaic trigonometric function, versine (versed sine), to overcome the precision problem of small angles when using the law of cosines (Sinnott, 1984). The versed sine of an angle is one minus the cosine of the angle. Half the versed sine is known as haversine (hav). The use of

$$\text{hav}\theta = (1 - \cos\theta) \div 2 = \sin^2(\theta \div 2),$$

the haversine formula below was derived by Sinnott to calculate the geodetic distance using standard trigonometric functions:

$$d = 2R \cdot \tan^{-1}2(\sqrt{a}, \sqrt{1-a})$$

where $a = \sin^2[(\phi_1 - \phi_2)/2] + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2[(\lambda_1 - \lambda_2)/2]$, with R, ϕ and λ defined as above.

The $\tan^{-1} 2(\sqrt{a}, \sqrt{1-a})$ function used here takes two arguments, $\sqrt{a}$ and $\sqrt{1-a}$. These two arguments are used instead of $\tan^{-1}(\sqrt{a}/\sqrt{1-a})$ to be able to handle $a = 1$ and circumvent division by zero. Using the two arguments also ensures that the function returns values in all four quadrants.

In the haversine formula, the inverse tangent is used to convert the angle (in radians) to a distance. The inverse tangent experiences hardly any loss in precision in this conversion for small angles, as the tangent of a small angle is approximately the value of the angle. Therefore, the haversine formula, rather than the law of cosines, should be used for distances of one metre or less (Huber, 2011).

The haversine formula is not flawless though. Sinnott (1984) pointed out that antipodal (points on the earth's surface which are diametrically opposite) or near antipodal points do not yield accurate results. Antipodal points are generally regarded as special and unusual cases, which require much more rigour when selecting an appropriate method to calculate geodetic distances when needed.

3.5.3 The ellipsoidal earth model

Using a more complicated model for the earth results in more complicated calculations when solving for geodetic distances. However, these calculations can be solved very quickly on modern computers.

Vincenty (1975) derived iterative solutions with nested equations to calculate geodetic distances using the more accurate ellipsoidal earth model. He derived an inverse solution to calculate the distance between two given points with accuracy within millimetres from the theoretical ellipsoid. A drawback of this is that his solution fails to converge for antipodal or near antipodal points. Although Vincenty provided a modification to his method, this sometimes requires an extraordinary number of iterations before convergence is obtained (Karney, 2013). The Vincenty formulae are computationally more intensive, but not much slower than using the great-circle distance formula (StackOverflow, 2016).

Adapting earlier geodetic methods to modern computers, Karney (2013) developed algorithms for the ellipsoidal earth model to calculate geodetic distances quickly and accurately. Karney's algorithms converge at all points.

3.5.4 Accuracy of geodetic distances used

In general, the accuracy of geodetic distances is quoted in terms of the theoretical earth model used. The spherical earth model yields errors in geodetic distances of up to 0,3% from the theoretical ellipsoid, depending on the position of the points (Huber, 2012). With Karney's algorithms, the accuracy of the calculated geodetic distances have an accuracy of nanometres with reference to the theoretical ellipsoidal value (Karney, 2013).

To determine which one of the geodetic distance formulae to use, the specific application, the software available for the application and the theoretical accuracy of the formulae should be considered. In this study, the great-circle formula could be used as the distances between the control points are not less than a metre. The haversine formula and Vincenty's method are also options as the control points are not antipodal or near antipodal. However, Karney's algorithms could be used with probably the greatest accuracy. In Sections 3.3.2 and 3.4.1, it was shown that, in distance calculations, accuracy to one metre is sufficient for this study. Although greater accuracy could be obtained for the geodetic distances, it is not required in this study. Therefore, any of the above-mentioned formulae could be used.

In the end, SAS software was chosen for the geodetic distance calculations, as it was convenient to do so with SAS already being used for solving the orienteering problems and calculating the shortest paths. In SAS, the GEODIST function, which is based on the Vincenty formulae, was used to calculate the geodetic distances (SAS, 2012). These calculated geodetic distances have been rounded off to the nearest metre to ensure that the three different types of distances all have the same accuracy. A subset of 54 geodetic distances (edges) is listed in Table E.1 on page 123.

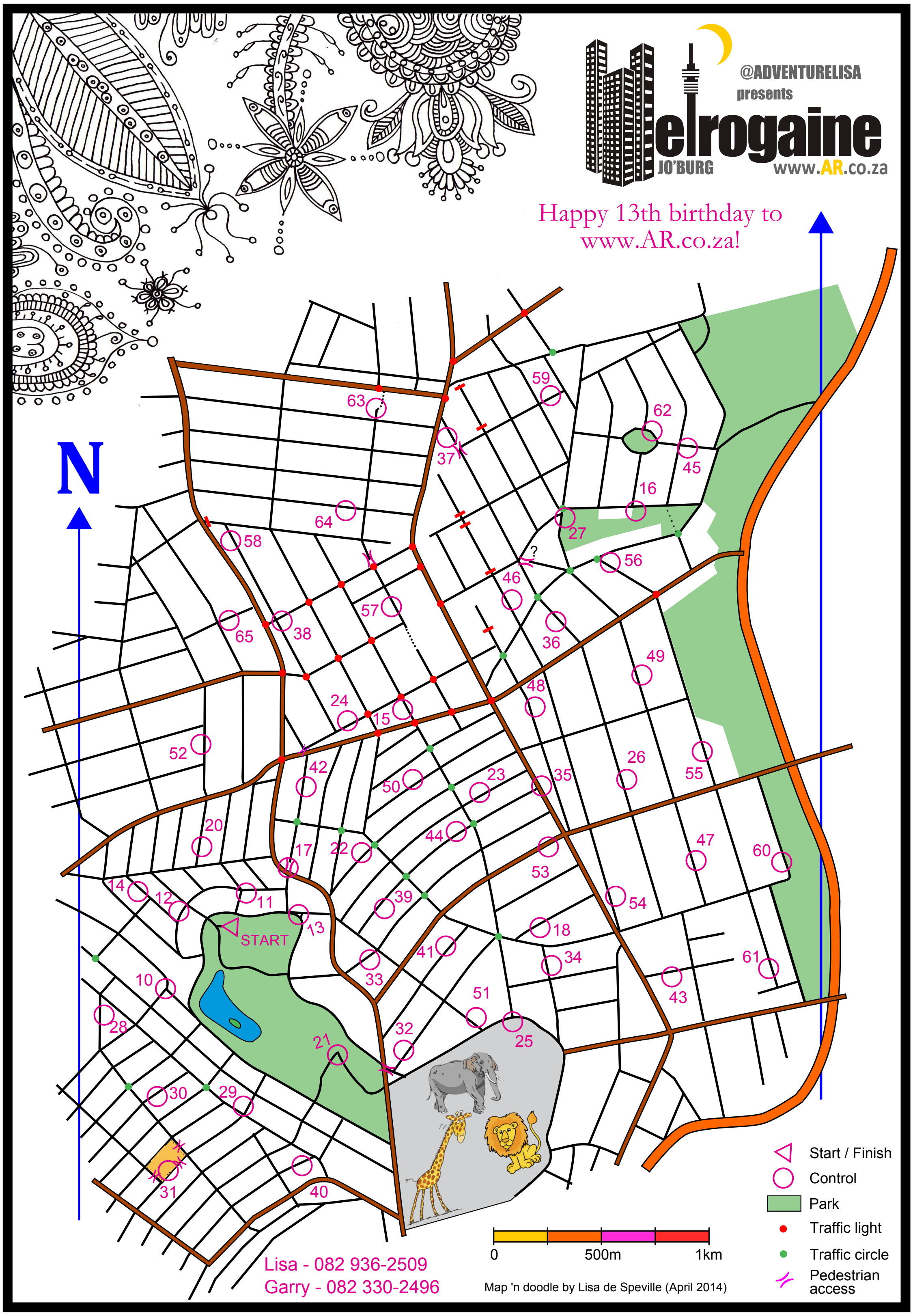
3.6 Conclusion

The data defined and described in this chapter is required as input for the orienteering problem and the different scoring approaches in Chapter 4. In terms of the graph model described in Section 3.1, three different sets of edges (distances) have been used in this study. In order to calculate these distances, the latitude, longitude and elevation of each control point were required. These data aspects were all addressed in this chapter and the results are summarised in Table 3.1 below.

	CP attributes	**Road edges**	**Elevation adjusted edges**	**Geodetic edges**
Described	Section 3.2	Section 3.3	Section 3.4	Section 3.5
Method	Maps	Dijkstra's	Naismith's	Vincenty's
Software	OSM; 1map	OSM; SAS	MS Excel	SAS
Applied	Distance calculations	Adjusted scoring Generated scoring OP	Generated scoring	Generated scoring
Accuracy	Lat: 1 m Lon: 1 m Alt: 10 cm	1 m	1 m	1 m
Data sets (elements)	Full set 56 Appendix C	Sub set 54 of 1 540 Appendix E	Sub set 54 of 1 540 Appendix E	Sub set 54 of 1 540 Appendix E

Table 3.1: Information on input data

With the distance data ready to be used in scoring the control points, the focus moves to the scoring approaches and algorithms, which are dealt with in the next chapter.



The Zoo Lake metrogaine map (De Speville, 2015)

Chapter 4

Scoring control points

This chapter deals with the scoring algorithms that lie at the core of this study. In the previous chapters the background was given, the methodology was discussed, the input data was prepared, and now the actual scoring of the metrogaine control points can begin.

The chapter begins by describing the current way of scoring control points. This is followed by a description of the scoring approaches that have been applied in this study. In addition, the different scoring algorithms flowing from the approaches are described and the new score sets, which are the end product of this chapter, are created.

The orienteering problem is formulated formally in this chapter. The OP input and output are discussed. In addition, benchmark score sets are introduced against which to compare the created score sets.

4.1 Current scoring approach

Currently, there are no set standards or rules for scoring control points in the orienteering community, merely guidelines. These guidelines (for score orienteering events) generally state that control points can be scored according to their distance and technical difficulty (British Orienteering, 2016), (Orienteering USA, 2015). The current approach to the scoring of metrogaine control points is described below.

During a metrogaine event, the teams should be constantly challenged in deciding which control point to go to next, and how to get there, in order to collect the maximum score points in the allotted time. When teams head off in different directions, it is indicative of an event where the control points are well scored, as scoring should be done in such a way that the teams spread out over the entire map area during the event. Teams must be challenged to make route choice decisions and not to follow obvious routes or subroutes. This is accomplished by having more control points on the map than the teams are able to visit, and by scoring the control points well.

Currently, the scoring of control points is a manual process. As previously mentioned, the factors involved in scoring a control point are the distance from the starting point, and the difficulty in reaching the control point. The distance is taken as the straight line distance between the start and a control point. The difficulty is measured in terms of the physical and navigational difficulty involved in getting to a control point.

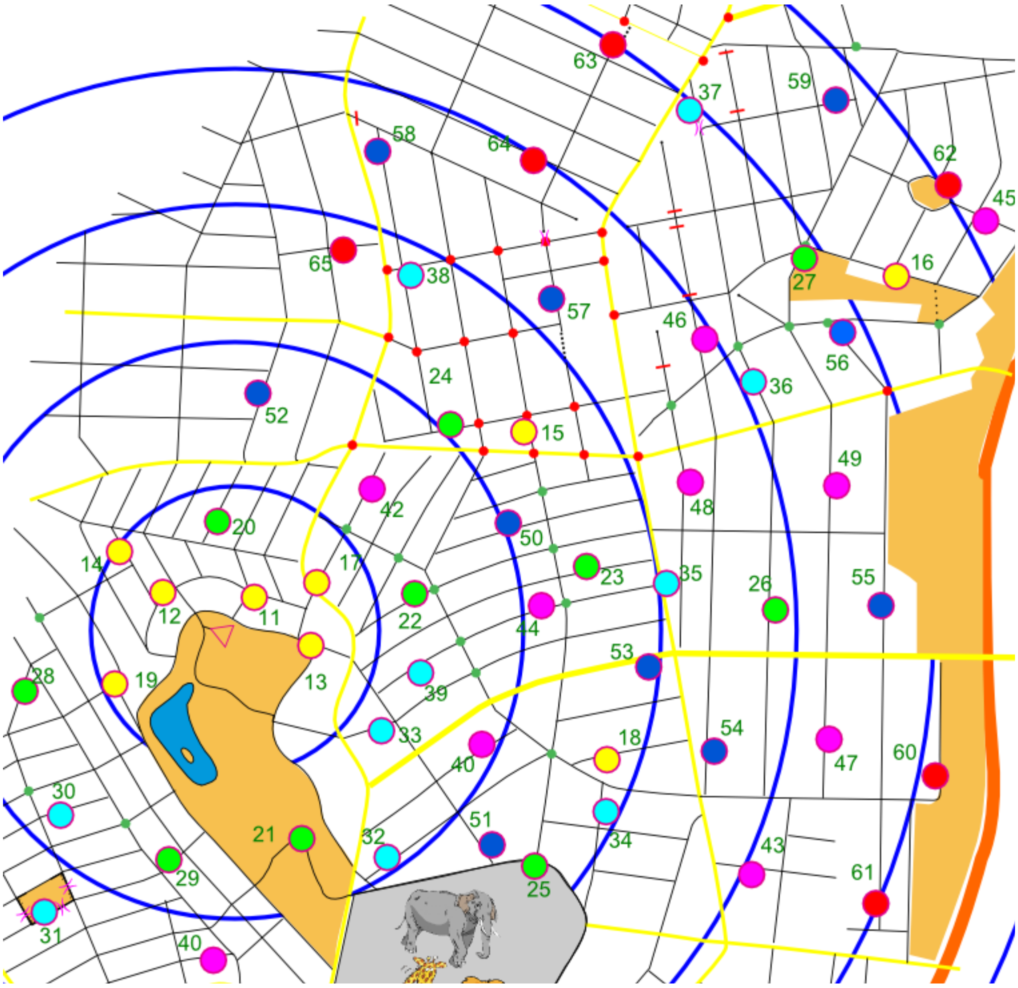


Figure 4.1: Zoo Lake map with concentric, equi-distance circles (De Speville, 2015)

The planner of the Zoo Lake metrogaine event has used concentric equi-distance circles, with the starting control point at the centre, to score the control points according to distance. The control points that are further away are scored higher than the ones closer to the starting point, while keeping the blocks of ten control points with the same score, in mind. (See Section 1.2 on page 8 for the block constraint originating from using the control point number to indicate its score.) The use of concentric equi-distance circles assists in identifying blocks of control points in the area between two concentric circles that could have the same score. In Figure 4.1, the 500 m equi-distance circles used by the planner are shown (De Speville, 2015). Different coloured control points are used to indicate their scores.

Distance is not the only criterion used in scoring the control points, as can be seen in Figure 4.1, where the control points in the same concentric circles are not all scored the same. This is only the first step in the scoring process. For the difficulty criterion, the planner has used her orienteering experience, as well as her knowledge of route planning and map topography, to score the control points. The planner described scoring some of the control points on the Zoo Lake map to the author as follows:

> Instead of putting too many weighted points at the far end I thought I'd add in some low scorers. The lower points far out are not far from high scorers and they're not far apart so it is easy to pick them up and to score a number of points but not too top heavy. I was hoping that this might also entice top runners going for big points to make more decisions – they could choose to leave some of the lower scorers far away. For example, do they go for 16 and 27 or just get 16 and shoot across to 56.

To score metrogaine control points according to distance only, is not difficult as distance can be quantified easily. However, incorporating the technical difficulty of a control point is much more challenging and each planner would do this differently and subjectively. The end result of scoring should be to send the teams off in many different directions and on many different routes during the event. Currently, this desired outcome for a well-scored event is not measured in any way. To be able to standardise the scoring process, scoring needs to be quantified and evaluated.

4.2 Introduction to scoring approaches

In this study, scoring approaches were developed as a quantitative tool to explore scoring control points consistently and objectively. Two simple scoring approaches were adopted as a basis for scoring metrogaine control points. The two approaches are the adjusting and generating scoring approaches, described in Sections 4.4 and 4.5, respectively. Several scoring algorithms have been developed or adapted using these approaches. The orienteering problem (OP) plays an integral part in evaluating these algorithms. Therefore, the OP is mathematically formulated below. The way in which the OP is used and fit into evaluating the scoring algorithms, unfolds in the sections that follow.

4.3 Solving the orienteering problem (OP)

The OP as described in the literature, has been discussed in Section 1.3.1 on page 10. In the chapter on methodology, the role of the OP as a tool for generating team routes was discussed in Section 2.5, starting on page 32, where it was indicated that SAS software would be used to solve the orienteering problems. In Section 3.1 on page 40, the metrogaine map with its control points, was described as an undirected, positive weighted graph. Solving the OP using SAS code, made use of the features of such a graph. One characteristic of a metrogaine event, namely, that the starting point is also the finishing point resulting in a cycle route, was also used in the SAS coding of the OP.

In the OP, the objective is to select the scored control points to be visited in such a way that the total score is maximised, subject to a distance constraint. In metrogaine events the time is set, resulting in a time constraint. By assuming a constant average speed, the distance constraint can be used as the equivalent of the time constraint. The mathematical OP formulation of Feillet et al. (2005) was adapted for metrogaine events to

$$\text{maximise} \quad \sum_{i=1}^{n} s_i y_i$$

subject to

$$
\begin{gathered}
\sum_{i=1}^{n} d_{ij}x_{ij} \leq d_{max}, \quad i,j = 1,2,\ldots,n \\
\sum_{j=1}^{n} x_{ij} = y_i, \quad i = 1,2,\ldots,n \\
\sum_{i=1}^{n} x_{ij} = y_j, \quad j = 1,2,\ldots,n \\
y_1 = 1, \\
\text{subtour elimination constraints,} \\
x_{ij} \in \{0,1\}, \quad i,j = 1,2,\ldots,n \\
y_i \in \{0,1\} \quad i = 1,2,\ldots,n
\end{gathered}
$$

where

$$
\begin{aligned}
n &= \text{total number of control points} \\
s_i &= \text{score of control point } i \\
d_{ij} &= \text{distance between control points } i \text{ and } j \\
d_{max} &= \text{maximum cycle distance} \\
y_i &= 1 \text{ if control point } i \text{ is selected; 0 otherwise} \\
x_{ij} &= 1 \text{ if the edge between control points } i \text{ and } j \text{ is visited; 0 otherwise.}
\end{aligned}
$$

The Zoo Lake metrogaine event as a case study is the focus of this study. The specific OP inputs for this event are listed below:

– The number of control points is 55, which excludes the start/finish.

– The different types of distances between all the control points for the Zoo Lake map, as determined in Sections 3.3, 3.4 and 3.5.

– The range of 19 selected distances, as described in Section 2.5, one distance at a time.

– The 26 score sets as described in the sections below, one set at a time.

For each score set, 19 orienteering problems were solved in SAS. The SAS code used for solving the orienteering problem in this study can be viewed in Appendix J, starting on page 148 (Pratt, 2015a; 2017). An example of the graphical OP solution is shown in Figure 4.2.

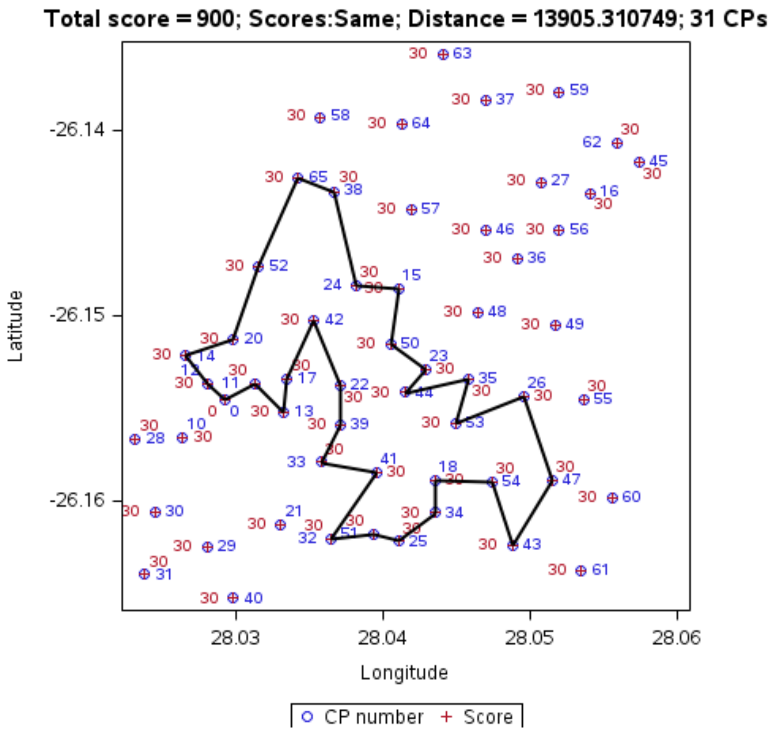


Figure 4.2: OP graphical output for `SAME` at 14 km

The OP outputs consist of the total maximum score, the number of control points selected, the control points selected, the order in which the control points have been visited and the cycle distance (in metres). Looking at Figure 4.2, the following should be noted:

- The heading shows the total score, the name of the score set used, the cycle distance in metres, and the number of control points visited (including the start/finish once).

- The small blue circles (with an enclosed red cross) show the actual geographical positions of all the Zoo Lake metrogaine control points in terms of their latitude (y axis) and longitude (x axis). (The convention is to write a latitude of 26 degrees south as -26 degrees.)

- The blue number adjacent to a circle indicates the control number, while the red number indicates the score.

- The control point number of the start/finish is 0 with a score of 0.

- The control points are connected with straight lines to simplify the graph, showing the order in which the selected control points have been visited. The

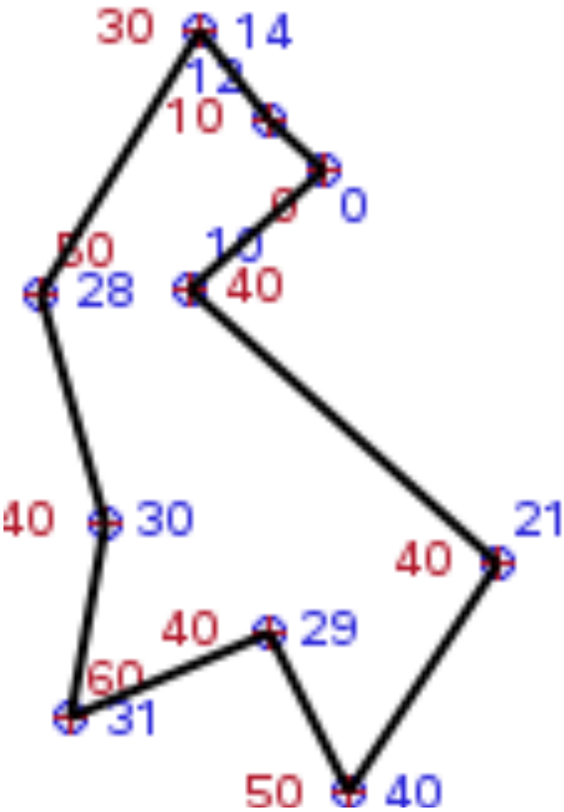


Figure 4.3: Simplified OP cycle

Figure 4.4: OP road cycle

length of a line segment between two control points is not equivalent to the road distance between these two control points.

Regarding the last listed item above, the graphical OP output from SAS for the score set `GeoOG100L` at 6 km, is shown in Figure 4.3, next to the actual OP road cycle for the same score set and distance in Figure 4.4. (The scale of the two figures differs.)

The time it took to solve one OP in SAS varied greatly. Some orienteering problems were solved in seconds, others took days, while others needed an intervention to obtain a solution. These problems, and how they have been overcome, are addressed in Section 6.2 on page 91. Although the progress of this study was negatively affected by these issues, in the end all the orienteering problems for all the score sets were solved.

4.4 Adjusting scoring approach

In the adjusting scoring approach, an initial score set is used to run the OP code for the 19 selected distances. In this study, only the road distances were used in the adjusting scoring approach, as these best model the distances covered by the teams best. Using the output of these 19 orienteering problems, the number of visits to each control point is summed over the 19 distances. These control point visit frequencies are then used to adjust the scores of the initial score set, using a chosen method.

In this study, the initial score set used in this approach, was the `SAME` score set. In the `SAME` score set, all the control point scores were given a value of 30. The actual score was not important as any multiple of ten below 70 would yield the same OP output. What was important was that the scores were all the same. These scores, together with the 19 distances identified in Section 2.5 on page 33, were used as input to solve the 19 orienteering problems with SAS code. From the SAS output, the control points selected to be visited for each of the 19 distances, were obtained. The numerical SAS output of each of the 19 orienteering problems was subsequently written to a spreadsheet in MS Excel for further calculations. The 19 graphical outputs using the `SAME` score set, are shown in Appendix K.1 on page 162. The graphical output for a distance of 14 km was shown earlier as an example in Figure 4.2.

In MS Excel, using `SAME`, the selected control points for the 19 distances were summed to give the frequency of visits to each of the 55 control points. These visit frequencies are shown graphically in Figure 4.5. The size of the bubbles represents the frequency, which is also written inside the bubble below the control point number. The centre point of each bubble is located at the latitude and longitude of the control point. The red dot in the bubble chart shows the position of the start/finish.

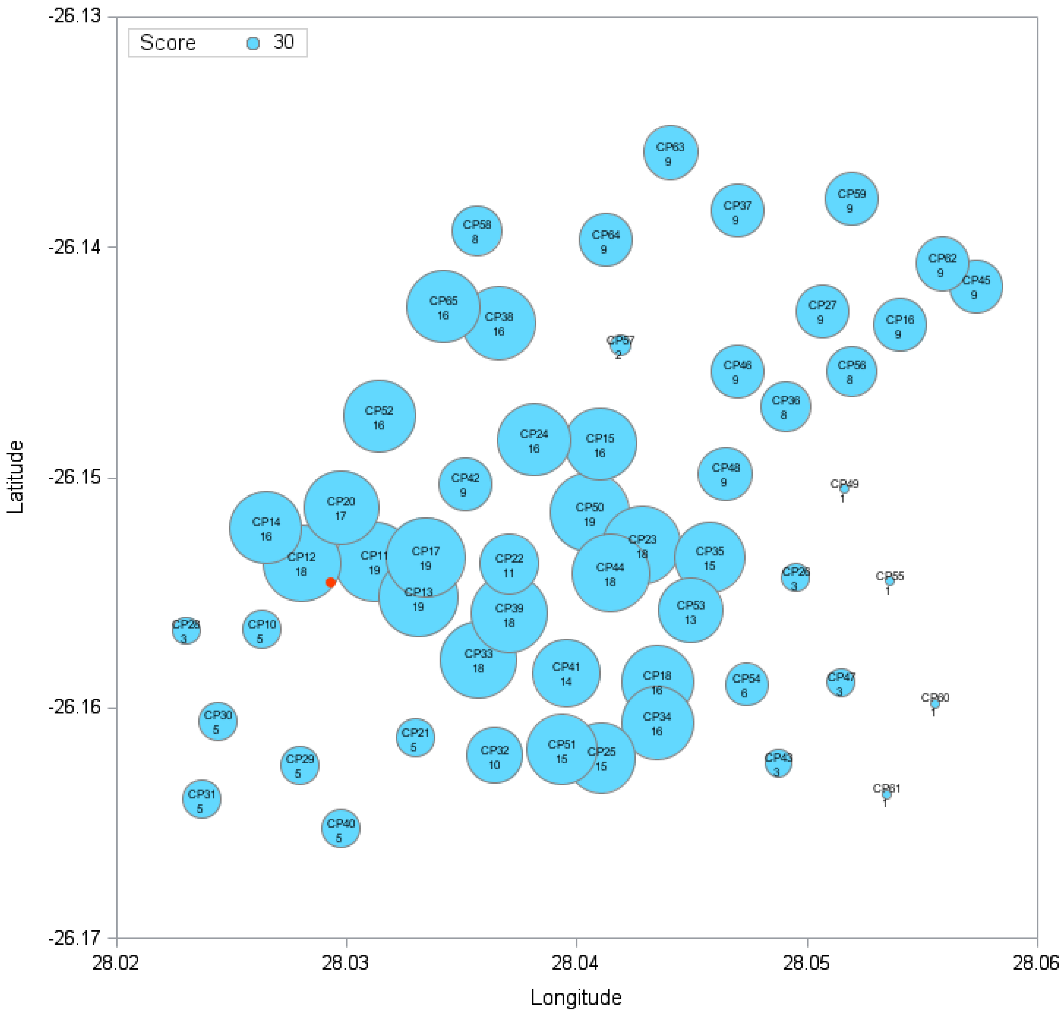


Figure 4.5: Visit frequencies resulting from `SAME`

From Figure 4.5, it can be seen that the two areas in the bottom corners of the bubble chart are the least visited, as indicated by the small bubbles. What is of more importance is that there are *areas* of low (and high) visitations.

To adjust the scores in `SAME`, the method chosen to adjust the control point scores is described. The 55 resultant visit frequencies were used to adjust the control point scores and the visit frequencies were then ordered from high to low. The scores of the ten control points with the highest number of visits were adjusted to 10, the scores of the ten control points with the next ten highest visits were adjusted to 20, and so on, until the scores of the five least visited control points were adjusted to 60. Ideally, all the control points should have the same number of visitations. By lowering the scores of the most visited control points (using `SAME`), these are made less attractive to visit, and vice versa. These blocks of ten control points, each having the same adjusted score, are due to the constraint originating from the way the score is indicated on the metrogaine map, as discussed in Section 1.2 on page 8. The resulting new score set was named SAME_I and can be viewed in Table I.1 on page 135.

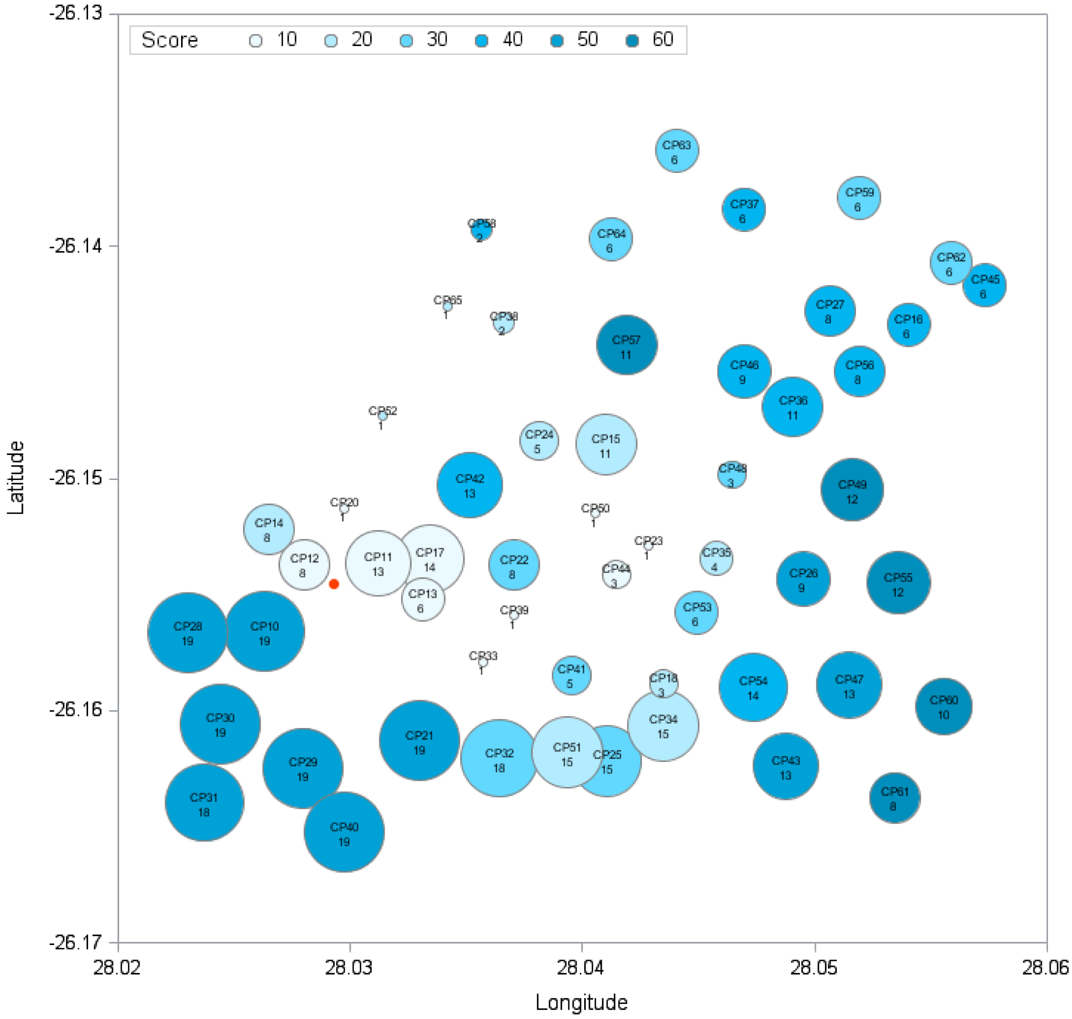


Figure 4.6: Visit frequencies resulting from SAME_I

In order to see the effect of the score changes, the summed visit frequencies for the new score set had to be determined. The SAME_I score set, together with the 19 selected road distances, were used as input to the 19 orienteering problems, and solved. The graphical output of the 19 SAME_I orienteering problems can be viewed in Appendix K.2 on page 165. The numerical output was written to a workbook in MS Excel to calculate the summed visit frequencies for each control point using the scores in `SAME_I`. In Figure 4.6, the bubble size shows the visit frequencies, while the different shades of blue show the different control point scores from the `SAME_I` score set. The score legend is located in the top left corner of the bubble chart.

In Figure 4.6, the two bottom corner areas of the bubble chart are now the most visited, as indicated by the large bubbles. From the two bubble charts above, it can be seen that the control point scores in a specific area are not independent; a specific control point is influenced by its neighbours, especially if they are all similarly scored. If control points in a specific area are all scored low, then teams will not choose to go to that area and the combined visitations will be low. However, if these control points were to be adjusted to all have high scores, then competitors would definitely choose to go to that area, resulting in high combined visitations for all control points in that area. Instead of levelling the visitations out by adjusting the scores, the visitations oscillated between very low and very high. Assigning the same score to ten control points, exacerbates the situation.

To avoid increasing (or decreasing) the scores of all the control points in a specific area, two methods were identified. The one method was to divide the control points into two groups geographically, adjusting the control point scores in the one group, but not in the other group. The other method was to determine an average index combining the number of visits and the inverse of the average number of visits of the nearest neighbours of a control point, where the inverse of the number of visits is defined as one divided by the number of visits. The application of these two methods is addressed below.

4.4.1 Adjusting by geographical groups

In order not to increase or decrease the scores of all of the control points in a specific area, all the control points were divided into two groups geographically, referred to for the sake of simplicity as the red and the blue group. The first attempt was to sort all the control points by latitude and assign every alternate control point to the same group. The result was not satisfactory as can be seen in Figure 4.7. The green marker on the map represents the start/finish.

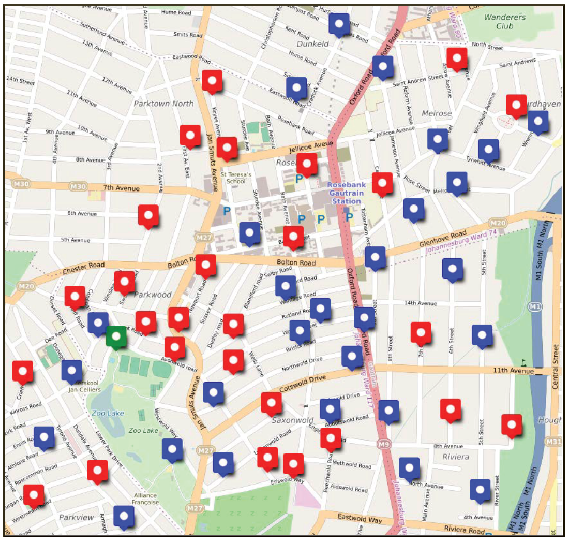


Figure 4.7: Groups by latitude

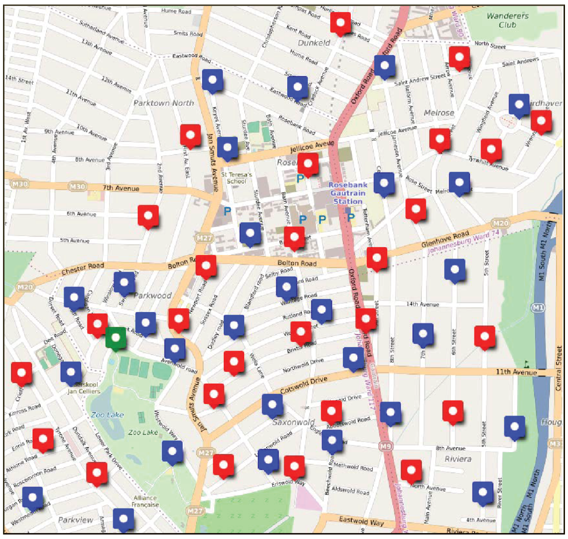


Figure 4.8: Groups by nearest neighbour

In the next attempt a control point and its nearest neighbour, by road, were assigned to different groups. It was possible to do this for 47 of the 55 control points. This method proved to be satisfactory as shown in Figure 4.8. Accordingly, the blue group comprised 28 control points and the red group 27.

The scores of the 27 red control points were kept at 30, while the `SAME` visit frequencies were used to adjust the scores of the blue control points. First, the visit frequencies of the blue control points were extracted and then ordered from the smallest to the biggest number of visits. Starting with the control point with the lowest number of visits at the top of the list, the scores of the first five blue control points were adjusted to 60, the next five to 50 and the next five to 40. No scores of 30 were assigned to any of the blue control points, as all the red control points were scored at 30. Thus, the scores of the next five blue control points were adjusted to 20. The scores of the remaining eight most visited blue control points were adjusted to 10. In Appendix I.1 on page 135, this new score set, `SAME_II`, can be viewed. It should be noted that the method described above, did not yield a score set (`SAME_II`) that adhered to the constraint of a maximum of ten control points with the same score.

4.4.2 Adjusting by frequency index

Another method chosen to address the dependence between the control point scores in a specific area was the use of a frequency index, where the visit frequencies and the inverse of the average visit frequencies of the nearest three neighbours of a control point were combined to adjust the scores. If the number of visits to a specific control point and its neighbours was high, the inverse of the number of visits would be low and vice versa. Therefore, combining the control point visit frequencies with the inverse of the neighbouring control points' visit frequencies should, in theory, result in a levelling-out process.

Using the shortest road distances determined in SAS, as described in Section 3.3.1 on page 46, the nearest three neighbours of each control point were identified. Subsequently, the average of the `SAME` visit frequencies of the nearest three neighbours of each control point was determined and inversed. To create the frequency index entry for each control point, the inverse (of the average of the visit frequencies to its three nearest neighbours) was added to its own visit frequency. These frequency index entries were ordered and used to adjust the scores in increasing multiples of ten as previously described, starting with the ten control points with the highest index, scored 10, and ending with the five control points with the lowest index, scored 60. The result is the score set, `SAME_III`, which can be viewed in Appendix I.1 on page 135.

4.5 Generating scoring approach

In the second scoring approach, a measure of work was defined and used to generate a score for each control point, as described below. The travelling salesman problem (TSP) was used as a basis in another generating scoring algorithm. Adaptations of the scoring approaches used by Tsiligirides and Fischetti complete the generating scoring algorithms used in this study.

4.5.1 Generating scores by measure of work

One way to describe the difficulty of getting to a control point is to look at the energy spent by a competitor (or team) in reaching the control point. The distance travelled on foot was taken to be a measure of the energy needed to reach a control point. In this study, the term “work” is used to describe the energy spent when moving on foot, using distance as an equivalent measure. Work was divided into global and local work. The measure of global work was taken as the distance a competitor had to travel from the starting point to the specific control point. Local work was defined as the marginal work that had to be done to get from one control point to the next, excluding the starting point. Three different types of distances were used as measures of global and local work, namely, geodetic distances, road distances and elevation adjusted road distances, which were discussed and calculated in the previous chapter. A score for each control point was generated by using these different measures of the work (distance) required to reach the control point.

The geodetic distances were calculated in SAS, using the latitude and longitude of each of the control points as described in Section 3.5 on page 49. The road distance is the actual distance travelled by road between control points, as described and determined in Section 3.3.1 on page 46. The elevation adjusted road distances were calculated using the road distances and the elevation differences between control points. For every metre of ascent moving from the starting point to a specific control point, ten metres of horizontal distance were added according to Naismith’s rule (Scarf, 2007). No correction was made in the case of descent. The measure of global work is defined as a directed edge (distance), always from the start to a specific control point. Not having to deal with undirected edges made the application of Naismith’s rule simple. The three types of distances, and the calculation thereof, were discussed in Chapter 3. A subset of 54 entries of the total 1 540 entries for the three different distances used in this study, can be viewed in Appendix E on page 123.

As defined above, the global work associated with a control point, is the distance travelled from the start to the control point. Three different measures of distance can be used as global work, namely, geodetic distances, road distances and elevation adjusted road distances. Therefore, the global work for all the control points is also a subset of the 1 540 edges that have already been calculated in the previous chapter, for these three types of distances. The 55 global work values for the three types of distances are available in Table F.1 on page 127.

Local work was defined above as the marginal work that had to be done to get from one control point to the next. The local work was taken as the average distance from the five nearest neighbours to a specific control point. Again, the three different types of distances (from the previous chapter) were used to calculate three measures of local work, which can be viewed in Table F.2 on page 129.

Regarding the calculation of the elevation adjusted road distances as measures for global and local work, the following should be noted. In Section 3.4 on page 48, Naismith's rule is used to give an equivalence between *ascent*, and horizontal distance, while making no adjustment for descent. In the case of global and local work, only directed edges (distances) are worked with. For global work, the work is calculated as the distance *from* the start *to* a specific control point. If the elevation of the control point is higher than that of the start, then the distance between the control point and the start is adjusted, otherwise not. In the case of local work, the work is calculated as the distance *from* its nearest neighbours *to* the specific control point. If the elevation of the control point is higher than that of its nearest neighbour, the distance between the control point and its neighbour is adjusted, otherwise not. Naismith's rule could thus be easily applied.

Different combinations of global and local work were used to generate control point scores. The global to local work ratios used were: 100:0, 80:20, 50:50, 20:80 and 0:100. The ratio of 100:0 represents 100% global work and no local work, while 0:100 takes 100% of the local work into account and none of the global work.

To generate the scores, the control points were ordered according to total work. The scores were allocated in increasing multiples of 10, starting with the ten control points with the *least* total work, and ending with the five control points with the most work, scored 60.

For each ratio of work, three distance types were used. The resultant score sets were named in line with the type of distance used, and the ratio between global and local work:

- Geodetic distances: `Geo100G0L`, `Geo80G20L`, `Geo50G50L`, `Geo20G80L` and `Geo0G100L`
- Road distances: `Road100G0L`, `Road80G20L`, `Road50G50L`, `Road20G80L` and `Road0G100L`
- Elevation adjusted road distances: `Alt_adj100G0L`, `Alt_adj80G20L`, `Alt_adj50G50L`, `Alt_adj20G80L` and `Alt_adj0G100L`.

The 15 score sets can be viewed in Appendix I.4, I.5 and I.6, starting on page 141.

4.5.2 TSP scoring algorithm

Based on the TSP, the shortest cycle when visiting all the control points once, was determined using SAS. The order of the control points visited in the TSP cycle was used to generate a score set. The control points closest to the start (five points on both sides) were scored the lowest at 10. Ten scores of 20 were assigned to the following five control points on each sides, moving away from the start in both directions. This assignment process was continued, each time increasing the score by 10, until the last five control points were scored 60. The `TSP` score set can be found in Table I.2 on page 137.

4.5.3 Tsiligirides scoring algorithm

Tsiligirides wanted to compare time standards in orienteering events to score standards in score orienteering events (SOE), as discussed in Section 1.3.7 on page 20. He wanted to change the scores in an SOE in order to make the time and score standards consistent across the two different formats of the sport.

The score scheme proposed by Tsiligirides (1984), created a new score for a specific control point i, by combining different ratios (r_1 and r_2) of two other scores ($S_1(i)$ and $S_2(i)$) generated for this specific control point. In general, Tsiligirides formulated the new score of control point i as $S(i) = r_1 \cdot S_1(i) + r_2 \cdot S_2(i)$, where $S_1(i)$ is the distance from control point i to its nearest neighbour, and $S_2(i)$ the average distance from control point i to all the other control points. In calculating $S_1(i)$ and $S_2(i)$, scaling factors were used to ensure that the new scores were within a practical range compared to the original scores. The new scores were rounded to the nearest five integer, a characteristic of the original scores. The r_1 and r_2 ratios used by Tsiligirides were 0,0:1,0; 1,0:0,5; 0,5:0,5 and 0,5:1,0.

Although Tsiligirides reported his approach as being unsuccessful, the incorporation of his approach in this study, with a different but similar aim to his, was done to see if a similar finding, or not, would be reached (Tsiligirides, 1984). The idea was not to repeat all of Tsiligirides' work, but to adapt his scheme and select some ratios to generate control point scores. As $S_1(i)$ and $S_2(i)$ are distances, they were named NN_d and Avg_d, respectively, in this study. Only three NN_d:Avg_d ratios were considered in the adapted Tsiligirides algorithm, namely, 0,0:1,0; 0,5:0,5 and 1,0:0,0.

Tsiligirides considered three score orienteering event problems where the location of the vertices were given as Cartesian coordinates and the edges (distances) were calculated as Euclidian distances. The distances he used to score a control point were all Euclidian distances. In the case of the Zoo Lake metrogaine event, the latitude and longitude coordinates were available, which point to geodetic distances (see Section 3.5). Road distances were chosen as the Tsiligirides distances, however, as they are more appropriate to use in an urban setting. The values of NN_d, the road distance to a control point's nearest neighbour, and Avg_d, the average road distance from a control point to all other control points, for all the control points were calculated in MS Excel, using the road distances from Section 3.3.1. These Tsiligirides distances are listed in Table G.1 on page 133.

Three ratios of the two distances were used to create the Tsiligirides score sets, namely, 0:100; 50:50 and 100:0. For each ratio, the resultant total distances for all the control points were calculated and then ordered. The ordered control points were used to allocate the scores in increasing multiples of 10, starting with the ten control points with the shortest total distance, and ending with the five control points with the longest total distance, scored 60. Three resultant score sets, `TS0Si100Sii`, `TS50Si50Sii` and `TS100Si0Sii` were generated and are available in Table C.3 on page 139.

4.5.4 Fischetti scoring algorithm

To conclude this section on generating approaches, the third scoring generation method used by Fischetti et al. (1998) was also incorporated. The first two methods (mentioned in Section 1.3.7 on page 21) have already been applied, resulting in the `SAME` and `RANDOM` score sets. This third score generation used by Fischetti and Toth (1988), generated a score by using a ratio of two distances, where both these distances were measured from the starting control point, but with different ending control points. The one distance was measured from the start to the specific control point (CP_d), and the other distance from the start to the control point furthest away from the start. For a specific map with its given, fixed control point positions, the latter distance would be a constant value. By multiplying the Fischetti ratio by 99 and adding 1, a score in the range [1; 100] is generated. An adapted Fischetti score generation approach was chosen as the final generating scoring algorithm used in this study.

The road distances were used again in this scoring algorithm, as these are the distances run by the teams in a metrogaine event. The road distances between all the control points were calculated in Section 3.3.1 on page 33, and used in this scoring algorithm. For the Zoo Lake metrogaine map, control point 16 (CP16) was the control point furthest away from the start at 3 388 m. The road distances from the start for all 55 control points were each then divided by 3 388 to yield the Fischetti ratio for each control point. See Table H.1 on page 133 for the distances (CP_d) and calculated Fischetti ratios. (Note that CP_d is the same as the global work using road distances.) Instead of generating scores the way Fischetti did, the method previously used by the author was applied to keep the scores as multiples of ten, and limit the number of control points with the same score to ten. Therefore, these Fischetti ratios were ordered from the smallest to the biggest. The first ten control points with the smallest ratios were scored 10, the next ten control points in the ordered list were scored 20, and so on, until the five control points with the biggest ratios were scored 60. The resultant score set, `FISCHETTI`, can be viewed in Table I.2 on page 137.

4.6 Benchmark score sets

In order to compare and evaluate the created score sets described above, two benchmark score sets, `EXISTING` and `RANDOM`, were introduced.

The `EXISTING` score set is the one that the planner of the Zoo Lake metrogaine map used when the event took place in April 2014. As explained previously, each control point score can be read from the unique, two-digit control point number (on the map), by multiplying the left digit by ten. The `EXISTING` score set is tabled in Appendix I.2 on page 137.

The second benchmark score set, `RANDOM`, was introduced to see if the results from the scoring algorithms fared better or worse than randomly generated control point scores. Although a series of randomly generated scores would probably have been better, only one was chosen for this study. A set of 55 random numbers was generated between 10 and 70, using the RANDBETWEEN function in MS Excel. These generated random numbers only had two digits. The left digit of each random number was multiplied by ten to give the 55 random scores. These random scores were kept in the order in which they were originally generated. To assign these random scores, the control points were sorted by latitude from the smallest to the largest value. In terms of the metrogaine map (Figure 3.1 on page 41), this means that the control points were in a sorted list, starting with the control points at the bottom of the map and ending with the control points at the top of the map. The 55 entries on the random score list were then assigned to the 55 entries on the sorted control point list, by entry number. That is, the first entry on the score list was assigned to the first entry of the control point list, the second score entry was assigned to the second control point entry, and so on. The score set, `RANDOM`, may be found in Table I.2 on page 137.

A comparison of the new score sets and the benchmark sets indicates whether the scoring algorithms do better or worse than the two selected benchmark standards. To add value, a scoring algorithm should fare better than the benchmarks.

4.7 Conclusion

In this chapter, seven different scoring algorithms were used to create 23 score sets, following two scoring approaches. The current scoring approach was also presented. Finally, the case for benchmarks was introduced and the two benchmark score sets that were selected for this study were discussed.

The culmination of the work done in this study, that is, the evaluation of the scoring sets created from different scoring algorithms, is presented in the next chapter.

Chapter 5

Evaluating scoring algorithms

In the previous chapter, different scoring algorithms were used to create 23 new control point score sets. To find out how well these different scoring algorithms fared in achieving the aim of spreading teams over the whole map in a metrogaine event, the score sets have to be evaluated. Therefore, in this chapter, the evaluation of the score sets take place, using the two scoring metrics developed in the second chapter. The chapter concludes with an analysis of the scoring metric results.

5.1 Introduction

To determine whether a scoring algorithm scores a metrogaine event well, the resulting score sets from the previous chapter should be evaluated. According to British Orienteering (2016), a well-scored event is one where teams will select many different routes, going in all directions over the map. This criterion was quantitatively modelled in Section 2.5 on page 32, and schematically presented in Figure 2.2. In this team spread model, a range of routes was generated by solving 19 orienteering problems for each one of the generated control point score sets. The number of visits to the selected control points were then summed over the 19 generated routes to give the visit frequency of each control point. These visit frequencies show how teams are spread over the map. Ideally, the visit frequency should be the same for all control points, with the teams spread evenly across the map. The deviation in the number of visits to a specific control point from the average frequency, gives an indication of how well the score set fared. The coefficient of variation (CV) and

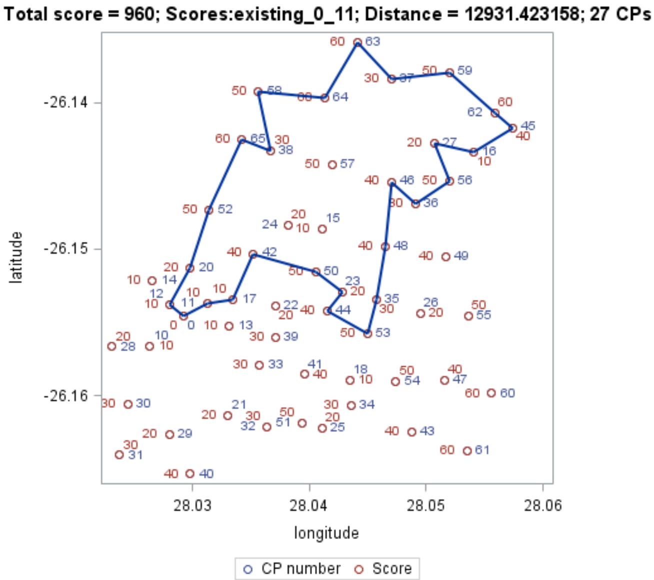


Figure 5.1: OP graphical output for EXIST at 13 km

the distribution uniformity (DU) are two quantitative measures used to determine how well a score set fared. See Section 2.6 on page 35 for a description of these two scoring metrics.

5.2 Spread

The first step to determine the spread resulting from a score set was to solve 19 orienteering problems using the specific score set, the range of 19 different route distances, and the road distances between all the control points, as input. (See Section 2.5 on page 33, and Section 3.3.1 on page 46 for information on the route distances and the road distances, respectively.) The graphical output for these orienteering problems, for each of the 23 created score sets, is available in Appendix K, starting on page 161. It should be noted, however that the graphical output for only 18 distances is shown for each score set in Appendix K. The route output for the travelling salesman problem (TSP) distance of 28,2 km will be the same for all score sets, as at this distance all control points are visited once. Therefore, the graphical orienteering problem (OP) output for 28,2 km is shown only once in Figure K.1 on page 161, although the OP using the TSP distance was solved for every score set and used for verification purposes. It is only the graphical output that is not shown to avoid unnecessary duplication.

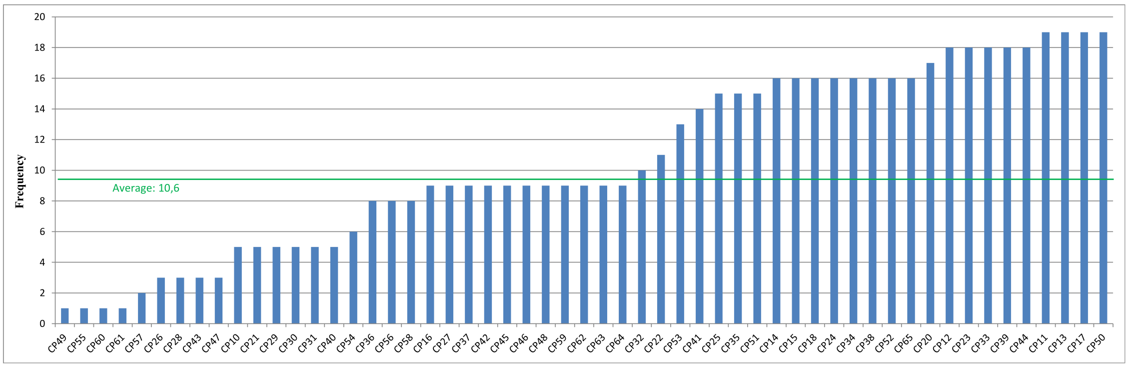


Figure 5.2: Ordered visit frequencies for `SAME`

An example of one of the 19 output graphs using the `EXIST` score set at 13 km is shown in Figure 5.1. Another example for the `SAME` score set at 14 km is shown in Figure 4.2 on page 60. For all OP output graphs, it should be noted that the lengths of the connecting lines do *not* show the distance length; they merely connect the control points to show the order in which the points have been visited.

The SAS output for each OP is presented in both graphical format (as in Figure 5.1) and numerical format. The numerical data was written to a separate MS Excel workbook for each score set. Each of the 26 workbooks contained 19 worksheets, one worksheet per route distance. The numerical data included the subset of control points visited, the order in which the control points were visited, and the maximum score points, subject to the route distance.

The second and final step in determining the spread resulting from a score set is to determine the total number of visits to each of the 55 control points for the specific score set. This was done by summing the visited control points from each OP output over the range of 19 distances in MS Excel. To present the spread, the ordered visit frequencies per control point for a specific score set were graphically displayed as a bar chart in MS Excel. In Figure 5.2, an example of such a bar chart for the `SAME` score set is shown. Each vertical bar shows the visit frequency determined over the range of 19 route distances.

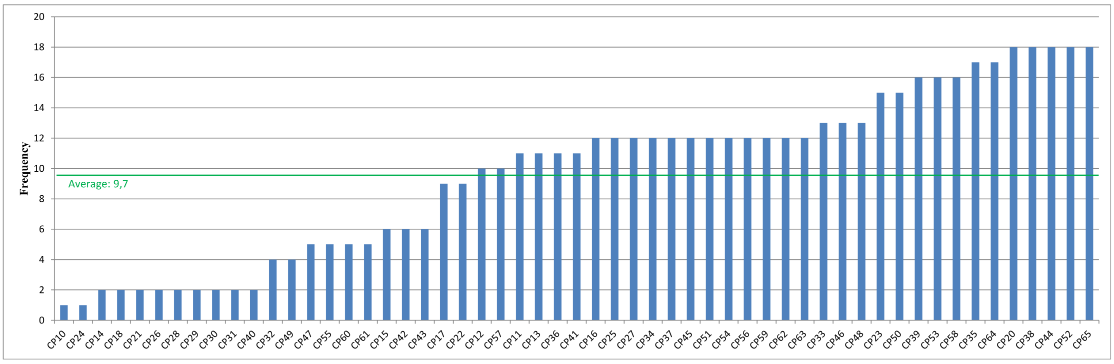


Figure 5.3: Ordered visit frequencies for EXIST

The average number of visits of 10,6 is shown on the chart as a horizontal line. Ideally, the spread (number of visits) should be the same for each control point. Therefore, the ideal spread would be the average number of visits. From the chart it can be seen by how much the visit frequency for each control point deviates from the ideal.

In Figure 5.3, another visit frequency bar chart is shown, this time for the EXIST score set. For this score set, the ideal spread is at 9,7 visits per control point.

Although the control points with the largest spread deviation, as well as the size of the deviation, can be easily identified on the bar chart, it does not show where these control points are located. To incorporate this information, bubble charts have been plotted for each score set.

In Figure 5.4, the bubble chart for the EXIST score set is shown. The size of the bubble indicates the visit frequency of the specific control point, which is also given inside the bubble, below the control point number. Ideally, the bubble sizes should all be the same, indicating that all the control points were visited equally. The centre of the bubble gives the longitude and latitude of the control point, that is, its geographical location, which can be read off from the x and y axes, respectively. The colour of the bubbles are in shades of blue, showing the score of the control point. The score legend in the top left corner of the chart shows the lightest shade of blue, representing a score of 10, up to the darkest blue with a score of 60. The start/finish position is shown by the red dot on the graph. Accordingly, from the bubble chart, the locations of control points can be observed and the scores of the control points can be read. The bubble chart for the SAME score set has already been shown on page 63.

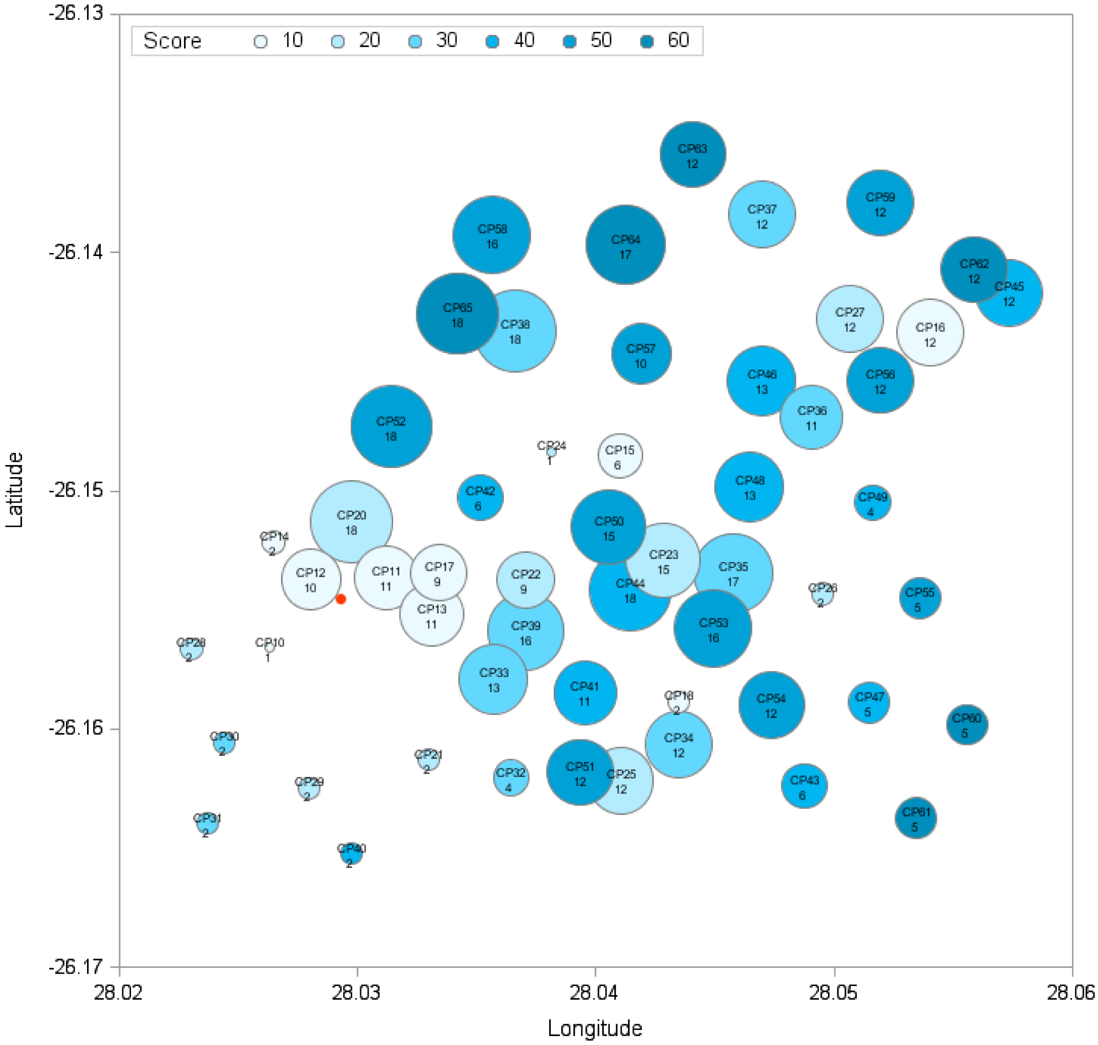


Figure 5.4: Visit frequencies resulting from EXIST

The bubble charts for all 26 score sets are shown in Appendix L, starting on page 241. From these bubble charts, the spread of the teams across the map can easily be visually observed.

Although a great deal of information can be obtained from the bubble charts, visually comparing the spread (the bubble sizes) of the different score sets is a challenge. Hence, such a comparison is best done quantitatively. Therefore, a quantitative scoring metric was needed to see how the different score sets fared and compared with each other. This quantitative measure is the topic of the next section.

5.3 Goodness of scoring

As discussed above, a scoring metric is needed to quantitatively evaluate how well a score set performs in terms of the spread of teams. To carry out this goodness of scoring evaluation, two test statistics, the coefficient of variation (CV) and the adapted distribution uniformity (DU), were defined and described in Section 2.6 on page 35. They were defined as

$$\mathrm{CV} = \frac{s}{Average\ frequency}$$

with

$$s = \sqrt{\frac{\sum (Observed\ frequency - Average\ frequency)^2}{n-1}},$$

where n is the number of control points, and frequency is the number of visits per control point, and

$$\mathrm{DU}_{\mathrm{UV}} - \mathrm{DU}_{\mathrm{LV}}$$

where

$$\mathrm{DU}_{\mathrm{LV}} = \frac{\text{Average of the lowest 25\% control point visits}}{\text{Average visits over all the control points}}$$

and

$$\mathrm{DU}_{\mathrm{UV}} = \frac{\text{Average of the highest 25\% control point visits}}{\text{Average visits over all the control points}}.$$

Using the visit frequencies, these two test statistics were calculated for the 26 score sets. The results are shown in Table 5.1, together with some descriptive statistics for all the score sets.

The goodness of scoring was quantitatively determined by calculating the test statistics, CV and ($\mathrm{DU}_{\mathrm{UV}} - \mathrm{DU}_{\mathrm{LV}}$), for all the data score sets. These goodness of scoring values will be analysed and discussed in the next section.

Data score set	**Total visits**	**Average**	**Standard deviation**	**CV**	$\mathbf{DU_{UV} - DU_{LV}}$
EXIST	533	9,691	5,4599	0,5634	1,4373
RANDOM	549	9,9818	5,7975	0,5808	1,4526
TSP	541	9,8364	6,5572	0,6666	1,7046
SAME	581	10,5640	5,8334	0,5522	1,2513
SAMEI	496	9,0182	5,7943	0,6425	1,6395
SAMEII	531	9,6545	6,3369	0,6564	1,6425
SAMEIII	516	9,3818	6,338	0,676	1,6978
Geo100G0L	521	9,4727	5,3881	0,5688	1,4327
Geo80G20L	522	9,4909	5,3778	0,5666	1,4299
Geo50G50L	528	9,6000	5,0574	0,5268	1,3616
Geo20G80L	516	9,3818	5,0753	0,5410	1,4009
Geo0G100L	502	9,1273	5,0883	0,5575	1,4478
Road100G0L	524	9,5273	5,4462	0,5716	1,4320
Road80G20L	523	9,5091	5,7053	0,6000	1,4798
Road50G50L	528	9,6000	5,1554	0,5370	1,3198
Road20G80L	510	9,2727	5,6420	0,6084	1,5252
Road0G100L	497	9,0364	4,6387	0,5133	1,2726
AltAdj100G0L	533	9,6909	6,0396	0,6232	1,5405
AltAdj80G20L	532	9,6727	5,9630	0,6165	1,5286
AltAdj50G50L	522	9,4909	6,0365	0,6360	1,5654
AltAdj20G80L	522	9,4909	5,0291	0,5299	1,3622
AltAdj0G100L	498	9,0545	4,7157	0,5208	1,2780
TS100SI0SII	479	8,7091	4,9354	0,5667	1,4599
TS50SI50SII	513	9,3273	5,3543	0,5740	1,4167
TS0SI100SII	523	9,5091	5,1778	0,5445	1,3296
FISCHETTI	525	9,5455	5,4496	0,5709	1,4218

Table 5.1: Descriptive and dispersion statistics of data score sets

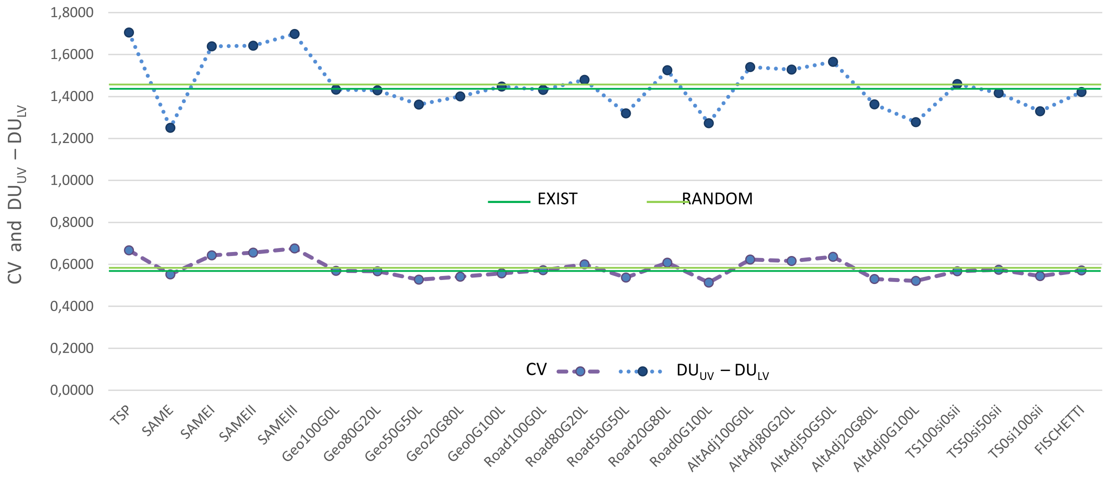


Figure 5.5: Scoring test statistics for the created score sets

5.4 Analysis of scoring metric results

In the previous section, the values of two test statistics, CV and $(\mathrm{DU_{UV}} - \mathrm{DU_{LV}})$, for all the data score sets were calculated and are summarised in Table 5.1. These values are now plotted for the 24 created score sets on one line chart in Figure 5.5, with the values of the two benchmark score sets drawn as horizontal lines. (Note that the dashed lines connecting the value points are just added to assist in the visual comparison of the two test statistics.) Ideally, both test statistics should approach zero, as discussed in Section 2.6. Therefore, the lower these test statistic values are, the better the performance of the score set.

When comparing the values of CV and $(\mathrm{DU_{UV}} - \mathrm{DU_{LV}})$ on the chart, it can be seen that the two test statistics generally follow a similar trend but do not give exactly the same results. When the CV value of a score set is smaller than the CV value of the benchmarks, the same is true for the $(\mathrm{DU_{UV}} - \mathrm{DU_{LV}})$ values, but not the other way round.

Looking at the CV values, nine score sets performed better than the `EXIST` benchmark score set, but 12 score sets performed better than `EXIST` when looking at the $(\mathrm{DU_{UV}} - \mathrm{DU_{LV}})$ values. This difference might be attributed to the fact that only 50% of the data control points were used in calculating the $(\mathrm{DU_{UV}} - \mathrm{DU_{LV}})$ values, while 100% of the data was used in calculating the CV value. Consequently, the CV value is chosen as the test statistic to be used in the following analysis.

Rank	Data score set	CV	Remark
1	Road0G100L	0,5133	
2	AltAdj0G100L	0,5208	
3	Geo50G50L	0,5268	
4	AltAdj20G80L	0,5299	
5	Road50G50L	0,5370	
6	Geo20G80L	0,5410	
7	TS0SI100SII	0,5445	
8	SAME	0,5522	
9	Geo0G100L	0,5575	
10	EXIST	0,5634	Benchmark
11	Geo80G20L	0,5666	
12	TS100SI0SII	0,5667	
13	Geo100G0L	0,5688	
14	FISCHETTI	0,5709	
15	Road100G0L	0,5716	
16	TS50SI50SII	0,5740	
17	RANDOM	0,5808	Benchmark
18	Road80G20L	0,6000	
19	Road20G80L	0,6084	
20	AltAdj80G20L	0,6165	
21	AltAdj100G0L	0,6232	
22	AltAdj50G50L	0,6360	
23	SAMEI	0,6425	
24	SAMEII	0,6564	
25	TSP	0,6666	
26	SAMEIII	0,6760	

Table 5.2: Ordered CV values for the data score sets

The CV values are ordered from the smallest to the biggest value in Table 5.2, with the two benchmark score sets marked in red for easy reference. Keeping in mind that the lower the CV value, the better the score set performed, the following observations are made:

- Sixteen of the 24 score sets performed better than the `RANDOM` benchmark score set, including the other benchmark score set `EXIST`. This means that in most cases, the scoring algorithms do better than when the control point scores are randomly selected.
- The `SAME` score set performed unexpectedly well by being ranked 8th, two places better than `EXIST`. This implies that by not scoring the control points at all adds more value to the scoring process than 18 other scoring algorithms.
- The subjective scoring by the Zoo Lake planner performed fairly well (ranked 10th), showing that at the Zoo Lake event, using her experience, she succeeded to a large extent in scoring the control points well.
- The adjusting scoring approach performed the worst. All three score sets resulting from the adjusted scoring algorithms (`SAMEI`, `SAMEII` and `SAMEIII`), were ranked in the last four places together with the `TSP` score set.
- The work score-generated algorithms, using the three different types of distances, produced mixed results. (See Section 4.5.1 on page 67 for a discussion of these algorithms.) The algorithms using only, or mostly, local work performed the best.
- The best performing algorithm is the road distance score-generated algorithm with a global to local work ratio of 0:100. This means using no global work, only local work to generate the control point score, fared the best. Next, performing only 1,5% worse, was the altitude-adjusted distance score-generated algorithm, also with a global to local work ratio of 0:100.
- There is a disparity in the results for the work score-generated algorithms. The most pronounced is for the elevation adjusted distance, where one algorithm was ranked second, another fourth, and the other three right down in the rankings at 20th, 21st and 22nd.

– Another unexpected result is the TS0SI100SII score set, ranked 7th. This is the Tsiligirides scoring algorithm, where only the average distance of a control point to all other control points was used to score the control points. See Section 4.5.3 on page 70 for the discussion on the Tsiligirides scoring algorithms. The two other Tsiligirides scoring algorithms were ranked 12th and 16th.

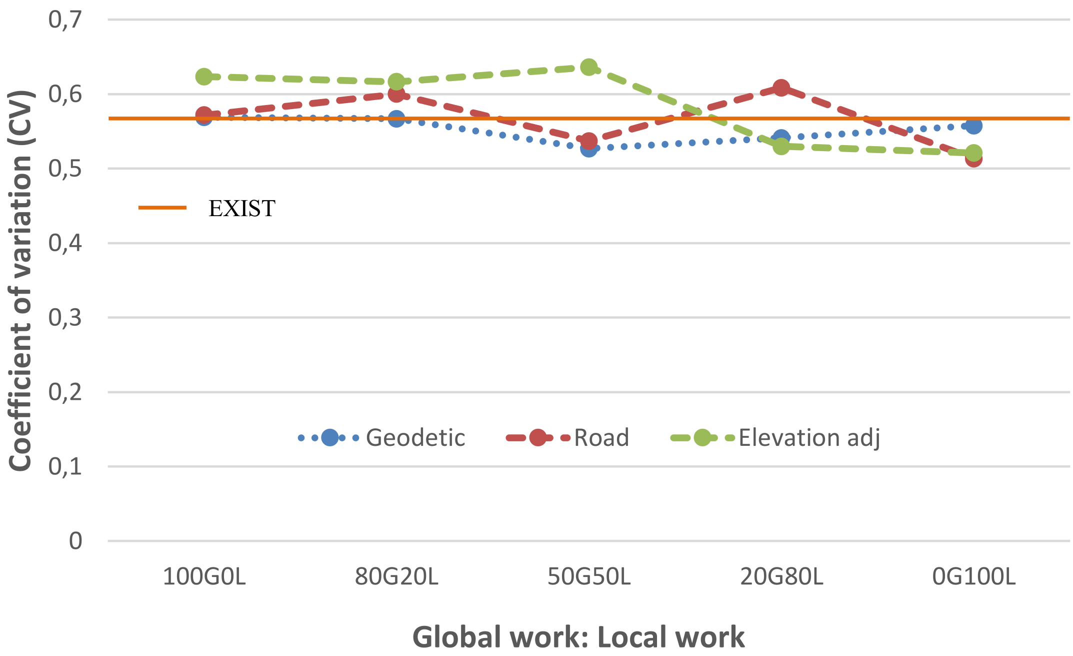


Figure 5.6: CV values for work score-generated algorithms

In Figure 5.6, the CV values of the 15 work score-generated algorithms are plotted. The dashed lines connecting the value points are added as a visual aid to distinguish between the three sets of values. In the graph, no trend can be seen between the different global to local work ratios in these score-generated algorithms. The big disparity in the performance when using the elevation adjusted distances can clearly be seen in the graph. It seems that global work, as a measure of the energy spent to get from the start to a control point, is not a good parameter in scoring control points, as all three algorithms for the ratio of 100% Global work : 0% Local work, performed worse than EXIST.

The score-generated algorithms for scoring the control points were based on distance variants. Contrary to what was expected, the scoring performance did not improve from using geodetic distances to road distances to elevation adjusted distances. The use of elevation adjusted distances was an attempt to model technical difficulty (by including elevation), and it was expected that elevation adjusted distances would score a control point better than road and geodetic distances. The use of road distances was expected to perform better than geodetic distances, as road distances are the actual distances covered by the teams. The CV results showed mixed results with no clear-cut distinctions between the distance types. Therefore, using a distance variant as the principle variable in scoring control points well was unsuccessful. The conclusion drawn from these CV result observations is that when scoring control points, other unidentified parameters play a more important role than distance variants.

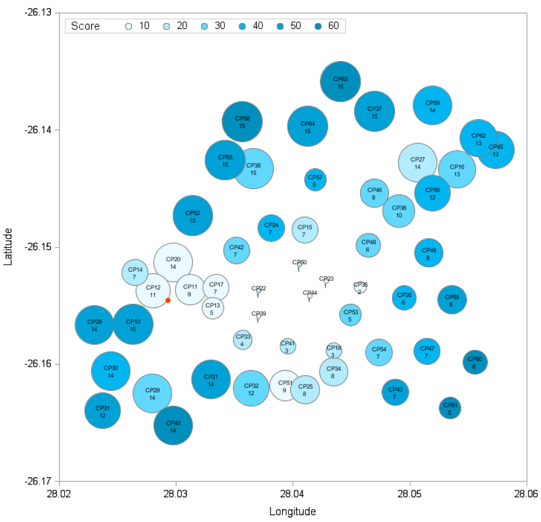


Figure 5.7: Bubble chart: `Road0G100L`

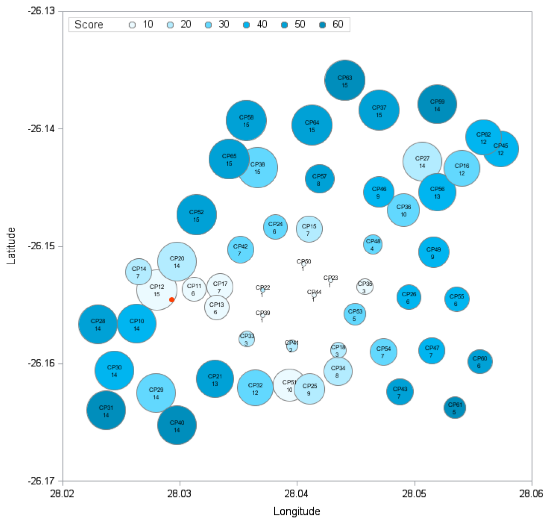


Figure 5.8: Bubble chart: `AltA0G100L`

Looking at the bubble charts of the two best performing scoring algorithms, `Road0G100L` and `Alt_adj0G100L` in Figures 5.7 and 5.8, it can be seen that there are areas with clusters of similarly scored control points. In areas where clusters of control point scores are high (darker blue bubbles), the visit frequencies (bubble size) to the cluster are also high, and vice versa. These clusters indicate that the interdependency between the scores of control points and their location in a specific area, has not been addressed in the score-generated algorithms.

The bigger CV values for the score-adjusted algorithms, in general, confirms that the dependency between the control points scores in a specific area of the map, could also not be addressed adequately by these algorithms. In these score-adjusted algorithms, the scores were adjusted using the visit frequencies from the `SAME` score set. Although attempts were made (see pages 65 to 67) to address the dependency issue, it is clear from the CV values of `SAMEII` and `SAMEIII` that these attempts were unsuccessful.

The analysis showed that the two scoring approaches did not yield the good, objective and sound scoring basis this research aimed to establish. Thus, quantifying technical difficulty in scoring control points by distance variants alone is not sufficient. Technical difficulty on its own will also not address the interdependency between the score and the location of neighbouring control points that was revealed in this study. It may therefore be concluded that simple quantitative scoring approaches such as those used in this study, are not designed to address the complex issues encountered in scoring control points.

5.5 Conclusion

In this chapter, the score sets created in the previous chapter were quantitatively evaluated. Two scoring test statistics developed in Chapter 2 were used in the evaluation. Although nine score sets did perform better than the benchmark score set, `EXIST`, the picture presented by the bubble charts for the scoring algorithms explored does not instil confidence in the use of the adjusted and generated scoring approaches to score control points well.

In the next and final chapter, a summary of the research findings is presented along with a section on possible future work that may be conducted.

Chapter 6

Conclusions and recommendations

In this final chapter, a summary of the research findings and the contributions of this study are presented. Problems that have been encountered are discussed before stating the conclusions of the research done. Recommendations for future research based on the findings, are also presented.

6.1 Summary of research

In Chapter 1 of this dissertation, the literature on score orienteering events and the related orienteering problem literature were reviewed. No study could be found in the literature that focused on scoring algorithms or the orienteering scoring problem, although two papers were found that treat the scoring of control points as a side issue. The focus of this study was described as exploring scoring algorithms in the sport of metrogaining, laying a foundation for a potential future scoring standard in the sport.

In the chapter on methodology, the two approaches selected for scoring metrogaine control points were discussed. The discussion on the adjusted and the generated scoring approaches led to the description of the different scoring algorithms chosen for this exploratory study. In the course of dealing with the methodology, the spread of teams was modelled to evaluate score sets. This model entailed generating a range of routes by solving a range of 19 orienteering problems. The visits to the control points on these generated routes were summed across the range of routes to describe the spread of teams across a metrogaine map for a specific score set. Two dispersion test statistics, the coefficient of variation and distribution uniformity, were adapted for the research problem to evaluate the created score sets. In both test statistics, spread was used to determine the goodness of scoring.

The data required as input for the orienteering problems and the different scoring algorithms were dealt with in a chapter on their own. Metrogaine map and OpenStreetMap (OSM) data were extracted, cleaned and formatted to be used as software input. Geodetic, road and elevation adjusted distances were defined and calculated using OSM, 1map, SAS and MS Excel.

The scoring approaches and algorithms formed the topic of Chapter 4. The current scoring approach was discussed and seven different scoring algorithms were used to create 23 score sets, using the adjusted and the generated scoring approaches. Two benchmark score sets, `EXIST` and `RANDOM`, were introduced for comparison with the created score sets. In the process of using the adjusted scoring approach to create specific scoring algorithms, a new scoring relationship emerged: the number of visits to a specific control point was dependent not only on its own score, but also on the scores of its neighbouring control points. This score dependency between neighbouring control points influences the number of visits to a control point. Changes were accordingly made to try and accommodate this dependency in two adjusted scoring algorithms. In 15 work-generated scoring algorithms, work was defined as the variable to be used in these algorithms. Work was subdivided into local and global work, using different types of distances to quantify the work.

In Chapter 5 of this study, the score sets created by using the different scoring algorithms were quantitatively evaluated to see how well they fared. Two scoring test statistics, CV and $(\mathrm{DU}_{\mathrm{UV}} - \mathrm{DU}_{\mathrm{LV}})$, were used in the evaluation. Using the CV value, seven score sets using the generated scoring approach performed better than the benchmark score set, `EXIST`, and eight score sets worse. These mixed results from the generated scoring approach indicate that the distance variants used in this approach were not able to score control points well. The adjusted scoring algorithms performed poorly, being ranked in the last four places according to their CV values. In addition, the two attempts to take the dependency between control point scores into account were unsuccessful. In the bubble charts, the score sets depicting the control point score and visit frequency for all control points, as well as the presence of score dependency, are clearly shown by the clusters of similarly scored control points. The findings of this study show that the adjusted and generated scoring approaches selected to score metrogaine control points, were not able to adequately describe the complex scoring process.

6.2 Discussion of problems

Three problems are to be addressed in this section. Two of the issues involve the orienteering problem (OP) in some way, and the last one deals with the block values constraint.

The first issue was the time to solve an OP. In this study, SAS software was chosen to solve the orienteering problems. The initial SAS code (`OrienteeringRowGen4`) used for solving the OP implemented a row generation approach by relaxing connectivity, while calling the mixed integer linear programming problem (MILP) solver multiple times (Pratt, 2015a). The `OrienteeringRowGen4` SAS code is available in Appendix J.2 on page 148. The time to solve one OP using this SAS code varied from seconds to hours, with a few not being solved in days. To overcome this problem, another SAS code, `OrienteeringFlow`, using a flow-based model in enforcing connectivity, while calling the MILP solver only once, was introduced (Pratt, 2015a). The second SAS code, `OrienteeringFlow`, is available in Appendix J.3 on page 152. When an OP running on one code took an excessive amount of time without finding an optimal solution, the program was stopped and the other code was used, which would then solve the problem within a reasonable time, sometimes within seconds. By alternating between the two SAS codes when required, the orienteering problems in this study could be solved. However, when using the `FISCHETTI` score set, there were five orienteering problems that ran for days on both SAS codes without a solution. A third SAS code, `OrienteeringRepair`, thus had to be sourced, which can be seen in Appendix J.4 on page 155 (Pratt, 2017). This code solves the connectivity relaxation by row generation when possible, and calls on the MILP solver only when the LP solution is connected. The code also tries to repair a disconnected integer solution by calling on the TSP (travelling salesman problem) solver. Using the `OrienteeringRepair` code, four of the five Fischetti orienteering problems were solved in minutes, while the fifth one (for the 9 km route) took around 3 hours.

The reason for the excessively long CPU times, without finding a solution in some cases, was not obvious and could not be established by the author. A consequence of these issues was that all 19 orienteering problems for one score set would sometimes solve within minutes, while others took several days, which had an impact on the progress of this study. The SAS code for solving the orienteering problems was merely used as a tool. To overcome the SAS time problem more elegantly, one needs to look deeper into the different codes and techniques to be more in control when solving these orienteering problems. A whole study on its own could be done looking at solving orienteering problems efficiently in SAS and other software for all route distances. Even solving orienteering problems with less than 100 control points has its challenges. This time problem was not pursued any further as it was not the focus of this study. In the end, all 494 orienteering problems were solved.

The second issue involving the OP revolves around multiple optimal solutions. In this study, once a solution was found for an OP, this first optimal solution was used. Although there are generally other optimal solutions for the same OP, they are not always easy to find or to know how many there are. In theory, it is possible to find all the optimal solutions for an OP by adding an outer loop to the SAS code that cuts off each solution as it is found, but this might not be feasible timewise (Pratt, 2015a). A number of optimal solutions per OP could be obtained and an optimal route consisting of the “average” visited control points be constructed for each OP, which would yield a different model from the one used in this study. The decision was taken to go with the simplified model in this study, using only the first optimal OP solution found for an OP.

The third and last issue deals with the block values constraint, where a maximum number of ten control points can have the same score. This constraint is a result of the way the control point scores are identified on the metrogaine map, using the left digit of the two-digit control point number in multiples of ten as the score. The limiting factor is that the second digit of the control point number only has ten different values. In this study, the current metrogaine practices were used, which include the maximum number of ten control points having the same score. However, in one score set, `SAMEII`, there were 27 control points with the same score of 30. It would therefore not be possible to use the current score identifying method with the `SAMEII` score set. The maximum of ten different digit values for the second digit can be overcome, without compromising recognisability, by replacing the second digit with one of the 26 letters of the alphabet (Swanepoel, 2017). If the second digit is retained, and replaced by a letter, the maximum number of control points with the same score increases to 36. This new maximum number would relax the block values constraint, while adapting the current practice without compromising the recognisability of the control point score on the map.

6.3 Conclusion

This study set out to explore scoring algorithms for metrogaine events. Two scoring approaches using simple distance type relations were selected to develop scoring algorithms. These approaches were selected by looking at the current manual approach, the available guidelines and two papers from the literature (Tsiligirides, 1984), (Fischetti et al., 1998). Twenty-three score sets were created from the scoring algorithms that were explored in this study. These score sets were quantitatively evaluated to determine how well they fared in terms of the spread model that was developed for this research.

In the process of exploring the scoring algorithms, it was assumed that the number of visits to a control point is dependent on the score of the control point. This is true, but what was also found is that this was not the only dependency. The number of visits to a control point is also dependent on the scores of its neighbouring control points. This dependency was observed when looking at the clustering of similarly scored control points in the visit frequency bubble charts (see Appendix L) drawn for the score sets. Whether this is the only other dependency at play in scoring algorithms, however, remains an unanswered question. What is known is that the factors affecting the scoring process are interdependent. Therefore, the use of just a distance-type variable in the scoring algorithms produced disparities in the goodness of scoring results. From these findings, the conclusion can be made that the scoring of control points is a much more complicated process than was originally envisaged. The exploration of scoring algorithms has consequently uncovered some of these complexities. In the end, the simple scoring approaches and algorithms explored in this study were unable to cope with these complexities. Hence, scoring approaches that are able to deal with complexity should be considered when scoring control points at metrogaine events.

6.4 Research contributions

In the process of exploring algorithms to quantitatively score control points in metrogaine events, the following research contributions were identified:

- Describing and defining the orienteering scoring problem.
- The development of a spread model for evaluating control point score sets.
- The introduction of two scoring test statistics for quantitatively evaluating goodness of scoring.
- The finding that at least some of the factors involved in scoring control points are interdependent.

6.5 Recommendation for future work

The relationship between the control point scores and the visit frequencies of the control points is complex, having interdependencies. Consequently, the simple approaches followed in this exploratory study were not able to describe the complexity and the interdependencies between the factors associated with the scoring process. Therefore, another scoring approach needs to be considered. Metaheuristics deal well with such complexities where the full extent of the complexity is not known or understood completely.

The orienteering scoring problem (OSP) is encoded below for general optimisation (Malan, 2017). This formulation could be used in future work involving metaheuristics.

$$\text{Minimise} \quad \text{CV of visit frequency}$$

subject to

$$\begin{gathered} 10 \leq i_1, \ldots, i_n < n + 10 \\ i_j \neq i_k \quad \forall \quad j \neq k \end{gathered}$$

where

$$
\begin{aligned}
\mathrm{CV} &= \text{coefficient of variation} \\
i &= \text{identifier of control point} \\
n &= \text{total number of control points}
\end{aligned}
$$

The OSP encoding is a way of specifying a solution in a form that can be searched by a metaheuristic. In the encoding above, the coefficient of variation of the control point visit frequencies must be minimised in the objective function. In the first constraint, the domain of the problem is set. The control point number is the unique identifier of a control point, with the first digit indicating the score of the control point in multiples of ten. The second constraint ensures that the control point identifier is unique. Thus, any sequence of control point numbers, subject to the two constraints, is a solution. Using a metaheuristic, solutions may be searched for to find an optimum for the OSP.

References

1map (2008–2017). Online version. `https://www.1map.co.za/`. Accessed: 2015–2017.

AdventureLisa (2011). Metrogaine: Nathan cleans up. `http://www.ar.co.za/2011/04/metrogaine-nathan-cleans-up/`. Accessed: 2012-10-25.

AdventureLisa (2011). Rogaining. `http://www.ar.co.za/2011/09/rogaining/`. Accessed: 2012-10-25.

Angelelli, E., Bazgan, C., Speranza, M. G., and Tuza, Z. (2014). Complexity and approximation for traveling salesman problems with profits. *Theoretical Computer Science*, 531:54–65.

Arkin, E. M., Mitchell, J. S. B., and Narasimhan, G. (1998). Resource-constrained geometric network optimization. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, SCG '98, pages 307–316, New York, NY. ACM.

Awerbuch, B., Azar, Y., Blum, A., and Vempala, S. (1998). New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. *SIAM Journal on Computing*, 28(1):254–262.

Bansal, N., Blum, A., Chawla, S., and Meyerson, A. (2004). Approximation algorithms for deadline-TSP and vehicle routing with time-windows. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 166–174, New York, NY, USA. ACM.

Biukaghazadeh, P. (2013). Basics of graph theory. In Farahani, R. and Miandoabchi, E., editors, *Graph theory for operations research and management: Applications in industrial engineering*, chapter 1, pages 1–13. IGI Global.

Blum, A., Chawla, S., Karger, D. R., Lane, T., Meyerson, A., and Minkoff, M. (2007). Approximation algorithms for orienteering and discounted-reward TSP. *SIAM Journal on Computing*, 37(2):653–670.

Bradley, T. (2007). *Essential statistics for economics, business and management*. John Wiley & Sons, Chichester, England.

Bratt, I. (2015). Private communication. Rand Orienteering Club.

British Orienteering (2016). Rules of orienteering. `https://www.britishorienteering.org.uk/page/rules`. Accessed: 2016-12-18.

Burt, C., Clemmens, A., Strelkoff, T., Solomon, K., Bliesner, R., Hardy, L., Howell, T., and Eisenhauer, D. (1997). Irrigation performance measures: efficiency and uniformity. *Journal of Irrigation and Drainage Engineering*, 123(6):423.

Butt, S. E. and Cavalier, T. M. (1994). A heuristic for the multiple tour maximum collection problem. *Computers and Operations Research*, 21(1):101 – 111.

Chamberlain, B. (2008). Calculating distances on the surface of the earth. `https://groups.google.com/forum/#!msg/comp.infosystems.gis/Yy14nK03oIU/moxwTGYbigEJ;context-place=forum/comp.infosystems.gis`. Accessed: 2017-07-20.

Chao, I.-M., Golden, B. L., and Wasil, E. A. (1996). The team orienteering problem. *European Journal of Operational Research*, 88(3):464–474.

Chekuri, C., Korula, N., and Pl, M. (2012). Improved algorithms for orienteering and related problems. *ACM Transactions on Algorithms*, 8(3):1–27.

Cherkassky, B. V., Goldberg, A. V., and Radzik, T. (1996). Shortest paths algorithms: Theory and experimental evaluation. *Mathematical Programming*, 73(2):129–174.

De Speville, L. (2015). Private communication. `lisa@adventurelisa.co.za`.

Dell'Amico, M., Maffioli, F., and Sciomachen, A. (1998). A Lagrangian heuristic for the prize collecting travelling salesman problem. *Annals of Operations Research*, 81:289–306.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Dye, S., Giffin, J., and Petty, N. (2010). A subset-selection prize-collecting TSP with uncertain speed. In *Proceedings of the 45th Annual Conference of the ORSNZ*, pages 336–344.

Eccles, D. W. (2008). Experts' circumvention of processing limitations: An example from the sport of orienteering. *Military Psychology*, 20(S1):S103–S121.

Feillet, D., Archetti, C., Hertz, A., and Speranza, M. G. (2009). The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*.

Feillet, D., Dejax, P., and Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation Science*, 39(2):188–205.

Fenton, D. (2001). Circumference at a given latitude. `http://mathforum.org/library/drmath/view/54158.html`. Accessed: 2017-11-16.

Fischetti, M., Gonzlez, J. J. S., and Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10(2):133–148.

Fischetti, M. and Toth, P. (1988). An additive approach for the optimal solution of the prize-collecting travelling salesman problem. In *Vechicle Routing: Methods and Studies*, volume 16, pages 319–343. Elsevier Science.

Gallo, G. and Pallottino, S. (1986). Shortest path methods: A unifying approach. In Gallo, G. and Sandi, C., editors, *Netflow at Pisa*, pages 38–64, Berlin, Heidelberg. Springer Berlin Heidelberg.

Garcia, A., Vansteenwegen, P., Arbelaitz, O., Souffriau, W., and Linaza, M. T. (2013). Integrating public transportation in personalised electronic tourist guides. *Computers and Operations Research*, 40(3):758 – 774. Transport Scheduling.

Geem, Z. W., Tseng, C.-L., and Park, Y. (2005). Harmony search for generalized orienteering problem: Best touring in China. In Wang, L., Chen, K., and Ong, Y. S., editors, *Advances in Natural Computation: First International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings, Part III*, pages 741–750, Berlin, Heidelberg. Springer Berlin Heidelberg.

Gendreau, M., Laporte, G., and Semet, F. (1998a). A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks*, 32(4):263–273.

Gendreau, M., Laporte, G., and Semet, F. (1998b). A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, 106(23):539–545.

Gensch, D. H. (1978). An industrial application of the traveling salesman's subtour problem. *AIIE Transactions*, 10(4):362–370.

Golden, B. L., Levy, L., and Vohra, R. (1987). The orienteering problem. *Naval Research Logistics*, 34(3):307–318.

Golden, B. L., Wang, Q., and Liu, L. (1988). A multifaceted heuristic for the orienteering problem. *Naval Research Logistics (NRL)*, 35(3):359–366.

Gordon, S. P. (2006). Rogaining: A prize-collecting orienteering problem. In *Proceedings of the 41st Annual Conference of the ORSNZ*, pages 173–182.

Grobbelaar, S. (2015). Private communication. `schalk@1map.co.za`.

Growcom (2013). Calculating distribution uniformity. `https://www.growcom.com.au/wp-content/uploads/2013/12/Calculating-distribution-uniformity-DU.pdf`. Accessed: 2017-06-21.

Gunawan, A., Lau, H. C., and Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332.

Gutin, G. and Punnen, A. P., editors (2002). *The traveling salesman problem and its variations*, volume 12 of *Combinatorial Optimization*. Kluwer Academic Publishers, Boston.

Hayes, M. and Norman, J. M. (1984). Dynamic programming in orienteering: Route choice and the siting of controls. *The Journal of the Operational Research Society*, 35(9):791–796.

Huber, W. (2011). Why is law of cosines more preferable than haversine when calculating distance between two latitude-longitude points? `https://gis.stackexchange.com/questions/4906/why-is-law-of-cosines-more-preferable-than-haversine-when-calculating-distance-b`. Accessed: 2016-07-20.

Huber, W. (2012). How accurate is approximating the earth as a sphere? `https://gis.stackexchange.com/questions/25494/how-accurate-is-approximating-the-earth-as-a-sphere#25580`. Accessed: 2017-07-20.

Ilhan, T., Iravani, S., and Daskin, M. (2008). The orienteering problem with stochastic profits. *IIE Transactions*, 40(4):406 – 421.

International Orienteering Federation (2015). About orienteering. `http://orienteering.org/about-orienteering/`. Accessed: 2015-07-26.

International Orienteering Federation (2017). International specification for control descriptions. `http://orienteering.org/resources/mapping/`. Accessed: 2017-10-25.

International Rogaining Federation (2012). The sport of rogaining. `http://www.rogaining.com/rogaining.html`. Accessed: 2012-10-26.

International Rogaining Federation (2015). What is rogaining? `https://www.rogaining.com/index.php?option=com_content&view=article&id=60:what-is-rogaining-more-detail&catid=27&Itemid=172`. Accessed: 2015-08-20.

IOF Map Commission (2017). International specification for orienteering maps. `http://orienteering.org/resources/mapping/international-specification-for-orienteering-maps-isom-2017/`. Accessed: 2017-07-14.

John Mittenthal, C. E. N. (1992). An insert/delete heuristic for the travelling salesman subset-tour problem with one additional constraint. *The Journal of the Operational Research Society*, 43(3):277–283.

Johnson, D. B. (1973). A note on Dijkstra's shortest path algorithm. *Journal of the ACM (JACM)*, 20(3):385–388.

Karney, C. F. F. (2013). Algorithms for geodesics. *Journal of Geodesy*, 87(1):43–55.

Kataoka, S. and Morito, S. (1988). An algorithm for single constraint maximum collection problem. *Journal of the Operations Research Society of Japan*, 31(4):515–530.

Kataoka, S., Yamada, T., and Morito, S. (1998). Minimum directed 1-subtree relaxation for score orienteering problem. *European Journal of Operational Research*, 104(1):139 – 153.

Keller, C. P. (1989). Algorithms to solve the orienteering problem: A comparison. *European Journal of Operational Research*, 41(2):224–231.

Laporte, G. and Martello, S. (1990). The selective travelling salesman problem. *Discrete Applied Mathematics*, 26(2-3):193–207.

Leibnitzer AC (2017). SO Cup 2017 map. 1:7 500. Retrieved from `http://omaps.worldofo.com/index.php?id=223264` on 2017-08-21.

Liang, Y.-C., Kulturel-Konak, S., and Smith, A. E. (2002). Meta heuristics for the orienteering problem. In *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC '02*, volume 1, pages 384–389.

Malan, K. (2017). Private communication. Decision Sciences, UNISA.

Moritz, H. (2000). Geodetic reference system 1980. *Journal of Geodesy*, 74(1):128–133.

Mouton, J. (2001). *How to succeed in your master's and doctoral studies: A South African guide and resource book*. Van Schaik, Pretoria.

OpenStreetMap contributors (2015). Version 0.8.2. `http://umap.openstreetmap.fr/en/`. Accessed: 2015–2017.

Orienteering USA (2015). Orienteering event types. `https://orienteeringusa.org/new-o/beginners-guide/event-types`. Accessed: 2015-10-24.

Ostrowski, K. and Koszelew, J. (2011). The comparison of genetic algorithms which solve orienteering problem using complete and incomplete graph. *Zeszyty Naukowe Politechniki Biaostockiej. Informatyka*, Z. 8:61–77.

Overpass turbo (2015). Web based filtering tool. `http://overpass-turbo.eu/`. Accessed: 2015-07-09.

Parise, C., Sternfeld, B., Samuels, S., and Tager, I. B. (2004). Brisk walking speed in older adults who walk for exercise. *Journal of the American Geriatrics Society*, 52(3):411–416.

Pekny, J., Miller, D., and McRae, G. (1990). An exact parallel algorithm for scheduling when production costs depend on consecutive system states. *Computers and Chemical Engineering*, 14(9):1009–1023.

Pekny, J. F. and Miller, D. L. (1992). A parallel branch and bound algorithm for solving large asymmetric traveling salesman problems. *Mathematical Programming*, 55(1-3):17–33.

Possum Trot Orienteering Club (2017). Shawnee Mission Park score map. 1:10 000. Retrieved from http://omaps.worldofo.com/index.php?id=196756 on 2017-08-21.

Pratt, R. (2015a). Private communication. SAS Advanced Analytics R&D, USA.

Pratt, R. (2015b). How to solve an orienteering problem? https://communities.sas.com/t5/Mathematical-Optimization/How-to-solve-an-orienteering-problem/m-p/229558. Accessed: 2015–2017.

Pratt, R. (2017). Private communication. SAS Advanced Analytics R&D, USA.

Quantico Orienteering Club (2014). About orienteering. http://qoc.us.orienteering.org/content/about-orienteering. Accessed: 2014-01-24.

RAC Orienteers (2017). Orienteering kite. [Photograph] Retrieved from http://www.racorienteers.co.za/ on 2017-12-01.

Ramesh, R. and Brown, K. M. (1991). An efficient four-phase heuristic for the generalized orienteering problem. *Computers & Operations Research*, 18(2):151–165.

Ramesh, R., Yoon, Y., and Karwan, M. H. (1992). An optimal algorithm for the orienteering tour problem. *ORSA Journal on Computing*, 4(2):155–165.

SAS (2002–2012). *Version 9.4*. SAS Institute Inc., Cary, North Carolina, USA.

SAS (2002–2015). *SAS 9.4 System Documentation*. SAS Institute Inc., Cary, North Carolina, USA.

Scarf, P. (2007). Route choice in mountain navigation, Naismith's rule, and the equivalence of distance and climb. *Journal of Sports Sciences*, 25(6):719–726.

Schilde, M., Doerner, K. F., Hartl, R. F., and Kiechle, G. (2009). Metaheuristics for the bi-objective orienteering problem. *Swarm Intelligence*, 3(3):179–201.

Sinnott, R. (1984). Astronomical computing: 2. Virtues of the haversine. *Sky and telescope*, 68(2):159.

Souffriau, W., Vansteenwegen, P., Vertommen, J., Berghe, G. V., and Oudheusden, D. V. (2008). A personalized tourist trip design algorithm for mobile tourist guides. *Applied Artificial Intelligence*, 22(10):964–985.

StackOverflow (2016). Is the haversine formula or the Vincenty's formula better for calculating distance? `https://stackoverflow.com/questions/38248046/is-the-haversine-formula-or-the-vincentys-formula-better-for-calculating-distance`. Accessed: 2017-07-21.

Swanepoel, C. (2017). Private communication. University of South Africa.

Sylejmani, K., Muhaxhiri, A., Dika, A., and Ahmedi, L. (2014). Solving tourist trip planning problem via a simulated annealing algorithm. In Biljanović, P., Butković, Z., Skala, K., and Golubić, S., editors, *Proceedings of the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2014)*, pages 1124–1129. IEEE.

Tasgetiren, M. F. and Smith, A. E. (2000). A genetic algorithm for the orienteering problem. In *Proceedings of the 2000 Congress on Evolutionary Computation.)*, volume 2, pages 910–915.

Thomadsen, T. and Stidsen, T. (2003). *The quadratic selective travelling salesman problem*. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark. IMM Technical Report.

Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *The Journal of the Operational Research Society*, 35(9):797–809.

uMap (2015). Online version. `http://umap.openstreetmap.fr/en/`. Accessed: 2015–2017.

Vansteenwegen, P., Souffriau, W., and Oudheusden, D. V. (2011). The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10.

Vansteenwegen, P. and Van Oudheusden, D. (2007). The mobile tourist guide: An OR opportunity. *OR Insight*, 20(3):21–27.

Velichko, S. (2004). Score-O. `http://www.ctoc-boise.org/ctoc/sergey/score-o.htm`. Accessed: 2012-10-19.

Vincenty, T. (1975). Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*, 23(176):88–93.

Wang, Q., Sun, X., Golden, B. L., and Jia, J. (1995). Using artificial neural networks to solve the orienteering problem. *Annals of Operations Research*, 61(1):111–120.

Wang, X., Golden, B. L., and Wasil, E. A. (2008). Using a genetic algorithm to solve the generalized orienteering problem. In Golden, B., Raghavan, S., and Wasil, E., editors, *The vehicle routing problem: Latest advances and new challenges*, pages 263–274, New York, NY. Springer Science+Business Media.

Wang, Y., Zou, Y., Henrickson, K., Wang, Y., Tang, J., and Park, B.-J. (2017). Google earth elevation data extraction and accuracy assessment for transportation applications. *PLOS ONE*, 12(4):1–17.

Wenig, J. (2010). Kosgei, Keitany shatter 25km world records in Berlin - Updated. *IAAF news story*. Accessed: 2017-10-23.

Wimberley, P. (2015). Private communication. RAC Orienteers.

Zhan, F. B. and Noon, C. E. (1998). Shortest path algorithms: An evaluation using real road networks. *Transportation Science*, 32(1):65–73.

Appendix A

Background information

A.1 Orienteering kite

Figure A.1: An orienteering kite (RAC Orienteers, 2017)

A.2 Metrogaine clue sheet

Page 1 of clue sheet:

TEAM

TOTAL POINTS: /2080

POINTS: (before penalties)

PENALTIES:

CATEGORY

RUN TIME:

COURSE (mark with an x)

60 min 90 min

NAME 1

NAME 2

10 points → Subtotal /40

10. What is outside house #84?
Statue Rock Fountain

11. Advertise here: call 011 514 ???? (On dustbin)
1400 2500 3600

12. Profession of owner at #74
Doctor Psychologist Dentist

13. How many large pots outside #7?
3 5 7

10 Points continued → Subtotal /40

14. What is the speed limit?(at #122 pedestrian gate)
20km/h 30km/h 40km/h

15. What is the sculpture at #59 doing??
Singing Sleeping Kissing

16. Bus stop sponsored by...
Walton's Bic Staedtler

17. Who protects #49?
Chubb ADT Best Secure

20 Points → Subtotal /160

20. What is inside #118?
Windmill Fountain Sculpture

21. 3 shapes on #87's gate?
Square Triangle Circles

22. Colour of post box roof. (#29)
Red Blue Green

23. What face is above the entrance of #9?
Moon Sun Planet

24. What colour are the machines at #73?
Orange Blue Grey

25. What is on top of pillars at #64?
Eagles Gargoyles Lions

26. ... Park? (Flats at #104)
Linton Linkin Dinton

27. Enter and be ...!!(#20 Pedestrian Gate)
Shot Prosecuted Eaten

30 Points → Subtotal /240

30. ... Security Group? (On guard house)
Eagle Hawk Colt

31. What is the huge plant outside #97?
Cactus Palm Willow

32. What colour is #187's post box?
Green Red Yellow

33. What is just inside the fence of the park?
Bench Guardhouse Playground

34. What nationality is the person on the sign at #81?
Mexican Italian Chinese

35. Name of house #141?
Bristol Briscoe Brisbane

36. How many lamps on #63's wall?
2 3 4

37. Fairland Graad R Sentrum opening time?
06:00 06:30 07:00

40 Points → Subtotal /160

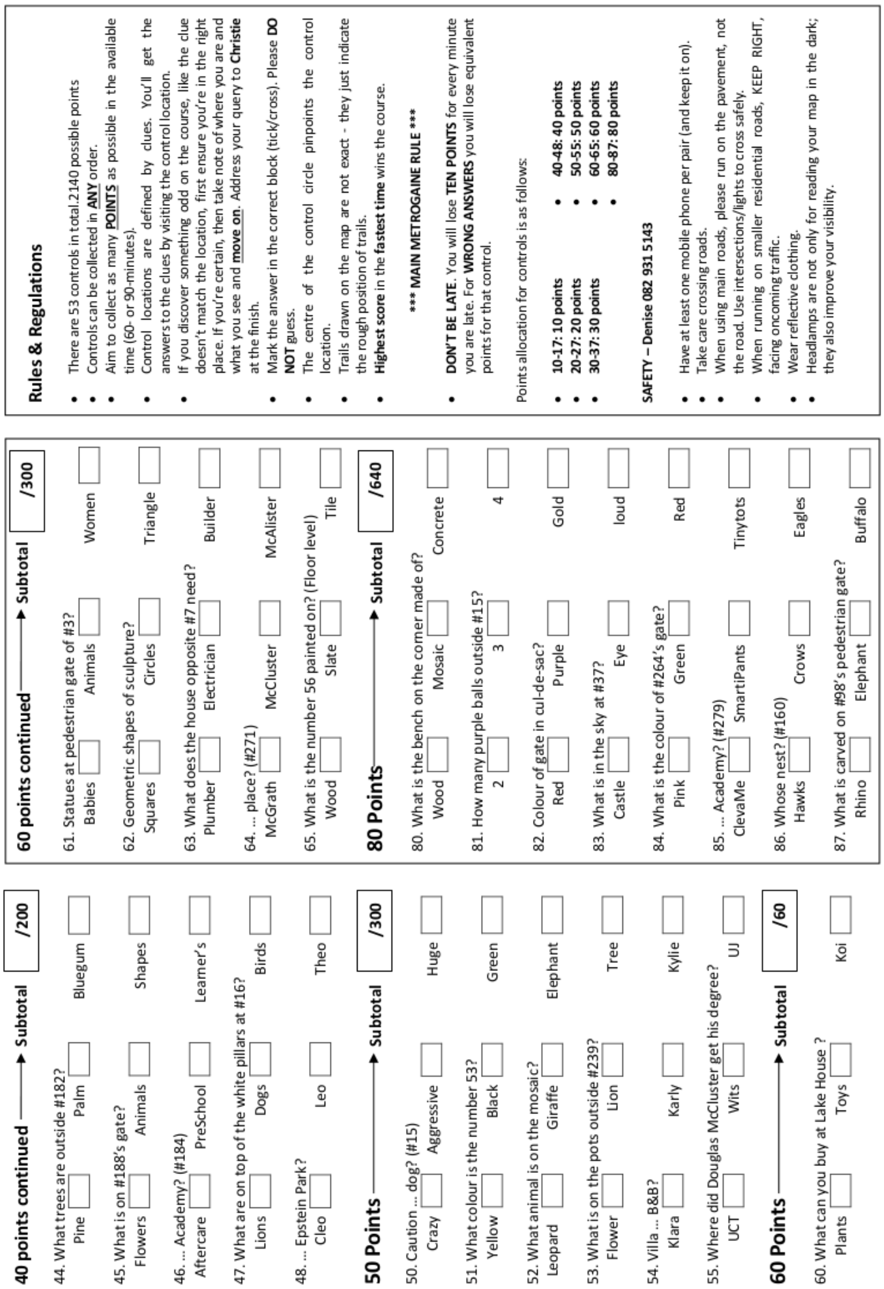
40. How many lights on the pole outside #134?
3 4 5

41. What is on the gate at #216??
Spear Angel Coat of Arms

42. ???? Flores lives at #175?
Mira Kira Tira

43. Which number is missing from #215's number? (Above Garage)
2 1 5

Page 2 of clue sheet:



40 points continued → Subtotal /200

44. What trees are outside #182?
Pine ☐ Palm ☐ Bluegum ☐

45. What is on #188's gate?
Flowers ☐ Animals ☐ Shapes ☐

46. ... Academy? (#184)
Aftercare ☐ PreSchool ☐ Learner's ☐

47. What are on top of the white pillars at #16?
Lions ☐ Dogs ☐ Birds ☐

48. ... Epstein Park?
Cleo ☐ Leo ☐ Theo ☐

50 Points → Subtotal /300

50. Caution ... dog? (#15)
Crazy ☐ Aggressive ☐ Huge ☐

51. What colour is the number 53?
Yellow ☐ Black ☐ Green ☐

52. What animal is on the mosaic?
Leopard ☐ Giraffe ☐ Elephant ☐

53. What is on the pots outside #239?
Flower ☐ Lion ☐ Tree ☐

54. Villa ... B&B?
Klara ☐ Karly ☐ Kylie ☐

55. Where did Douglas McCluster get his degree?
UCT ☐ Wits ☐ UJ ☐

60 Points → Subtotal /60

60. What can you buy at Lake House ?
Plants ☐ Toys ☐ Koi ☐

60 points continued → Subtotal /300

61. Statues at pedestrian gate of #3?
Babies ☐ Animals ☐ Women ☐

62. Geometric shapes of sculpture?
Squares ☐ Circles ☐ Triangle ☐

63. What does the house opposite #7 need?
Plumber ☐ Electrician ☐ Builder ☐

64. ... place? (#271)
McGrath ☐ McCluster ☐ McAlister ☐

65. What is the number 56 painted on? (Floor level)
Wood ☐ Slate ☐ Tile ☐

80 Points → Subtotal /640

80. What is the bench on the corner made of?
Wood ☐ Mosaic ☐ Concrete ☐

81. How many purple balls outside #15?
2 ☐ 3 ☐ 4 ☐

82. Colour of gate in cul-de-sac?
Red ☐ Purple ☐ Gold ☐

83. What is in the sky at #37?
Castle ☐ Eye ☐ loud ☐

84. What is the colour of #264's gate?
Pink ☐ Green ☐ Red ☐

85. ... Academy? (#279)
ClevaMe ☐ SmartiPants ☐ Tinytots ☐

86. Whose nest? (#160)
Hawks ☐ Crows ☐ Eagles ☐

87. What is carved on #98's pedestrian gate?
Rhino ☐ Elephant ☐ Buffalo ☐

Rules & Regulations

- There are 53 controls in total.2140 possible points
- Controls can be collected in **ANY** order.
- Aim to collect as many **POINTS** as possible in the available time (60- or 90-minutes).
- Control locations are defined by clues. You'll get the answers to the clues by visiting the control location.
- If you discover something odd on the course, like the clue doesn't match the location, first ensure you're in the right place. If you're certain, then take note of where you are and what you see and **move on**. Address your query to **Christie** at the finish.
- Mark the answer in the correct block (tick/cross). Please **DO NOT** guess.
- The centre of the control circle pinpoints the control location.
- Trails drawn on the map are not exact - they just indicate the rough position of trails.
- **Highest score** in the **fastest time** wins the course.

***** MAIN METROGAINE RULE *****

- **DON'T BE LATE**. You will lose **TEN POINTS** for every minute you are late. For **WRONG ANSWERS** you will lose equivalent points for that control.

Points allocation for controls is as follows:

- **10-17: 10 points**
- **20-27: 20 points**
- **30-37: 30 points**
- **40-48: 40 points**
- **50-55: 50 points**
- **60-65: 60 points**
- **80-87: 80 points**

SAFETY – Denise 082 931 5143

- Have at least one mobile phone per pair (and keep it on).
- Take care crossing roads.
- When using main roads, please run on the pavement, not the road. Use intersections/lights to cross safely.
- When running on smaller residential roads, KEEP RIGHT, facing oncoming traffic.
- Wear reflective clothing.
- Headlamps are not only for reading your map in the dark; they also improve your visibility.

Appendix B

Ethics approval

UNISA university of south africa

27 May 2014

Ref #: 2014_CEMS_SES_002

SCHOOL OF ECONOMIC SCIENCES
RESEARCH ETHICS REVIEW COMMITTEE

This is to certify that the application for ethics clearance submitted by

Mrs WA van Hoepen (student # 4648307, vhoepwa@unisa.ac.za)

Exploring algorithms to score control points in metrogaine events **received Ethics Approval**

The revised application for ethics clearance for the above mentioned research was reviewed by the School of Economic Sciences on 27 May 2014 in compliance with the Unisa Policy on Research Ethics. Ethical Clearance is granted.

You may proceed with the research project on condition that all participants are provided with Informed Consent forms prior to any fieldwork. Participation is strictly voluntary. The research ethics principles outlined by the Unisa Policy on Research Ethics must be adhered to throughout the project. Please be advised that the committee needs to be informed should any part of the research methodology as outlined in the Ethics application (Ref # 2014_CEMS_SES_002) change in any way or in case of adverse events. This certificate is valid for the duration of the project. The SES Research Ethics Review Committee wishes you all the best with this research undertaking.

Kind regards,

Ms C Loedolff
Chairperson of SES, CEMS, UNISA

Prof VA Clapper
Executive Dean: CEMS

Appendix C

Control point attributes

Table C.1: The Zoo Lake control point attributes

Control point	Latitude (degrees)	Longitude (degrees)	Elevation (m)
Start/Finish	−26.15458	28.02927	1 622.7
CP10	−26,15662	28,02631	1 614,0
CP11	−26,15368	28,03124	1 637,3
CP12	−26,15374	28,02807	1 617,8
CP13	−26,15524	28,03315	1 642,7
CP14	−26,15220	28,02650	1 609,6
CP15	−26,14856	28,04105	1 678,7
CP16	−26,14340	28,05407	1 640,3
CP17	−26,15347	28,03343	1 648,5
CP18	−26,15890	28,04351	1 675,7
CP20	−26,15133	28,02978	1 628,9
CP21	−26,16132	28,03301	1 637,4
CP22	−26,15378	28,03707	1 665,6
CP23	−26,15295	28,04284	1 682,7
CP24	−26,14840	28,03815	1 658,7
CP25	−26,16220	28,04106	1 676,7
CP26	−26,15436	28,04953	1 663,6
CP27	−26,41281	28,05069	1 655,4
CP28	−26,15667	28,02305	1 619,1
CP29	−26,16250	28,02799	1 634,7
CP30	−26,16061	28,02444	1 638,4

Table C.1 (*continued*)

Control point	**Latitude (degrees)**	**Longitude (degrees)**	**Elevation (m)**
CP31	−26,16400	28,02375	1 649,3
CP32	−26,16209	28,03646	1 656,3
CP33	−26,15794	28,03572	1 654,7
CP34	−26,16067	28,04352	1 678,6
CP35	−26,15348	28,04579	1 679,0
CP36	−26,14695	28,04907	1 670,6
CP37	−26,13843	28,04699	1 689,4
CP38	−26,14332	28,03663	1 662,2
CP39	−26,15594	28,03709	1 662,2
CP40	−26,16527	28,02977	1 654,0
CP41	−26,15851	28,03958	1 670,2
CP42	−26,15032	28,03519	1 645,0
CP43	−26,16243	28,04877	1 678,2
CP44	−26,15417	28,04149	1 679,6
CP45	−26,14172	28,05735	1 636,0
CP46	−26,14542	28,04698	1 676,2
CP47	−26,15893	28,05149	1 664,6
CP48	−26,14987	28,04646	1 681,6
CP49	−26,15053	28,05165	1 660,6
CP50	−26,15155	28,04056	1 677,0
CP51	−26,16184	28,03936	1 671,7
CP52	−26,14735	28,03145	1 631,2
CP53	−26,15581	28,04494	1 675,1
CP54	−26,15902	28,04740	1 663,2
CP55	−26,15453	28,05363	1 649,6
CP56	−26,14540	28,05196	1 652,1
CP57	−26,14428	28,04194	1 687,1
CP58	−26,13930	28,03566	1 663,2
CP59	−26,13794	28,05195	1 669,5
CP60	−26,15985	28,05560	1 659,2
CP61	−26,16377	28,05346	1 679,7
CP62	−26,14075	28,05590	1 646,1
CP63	−26,13591	28,04410	1 698,0
CP64	−26,13971	28,04127	1 686,8
CP65	−26,14259	28,03421	1 646,3

Appendix D

OpenStreetMap (OSM) data

D.1 Extracting the data

The processes followed to extract data from maps, and to clean and format the extracted data used in this study, are described below.

All road distances on the Zoo Lake map were obtained from OSM. Looking at a map in OSM, there is a great deal of data in the background which defines exactly how the map is drawn. The first step was to export the OSM map data for the specific area, using the export window as shown in Figure D.1.

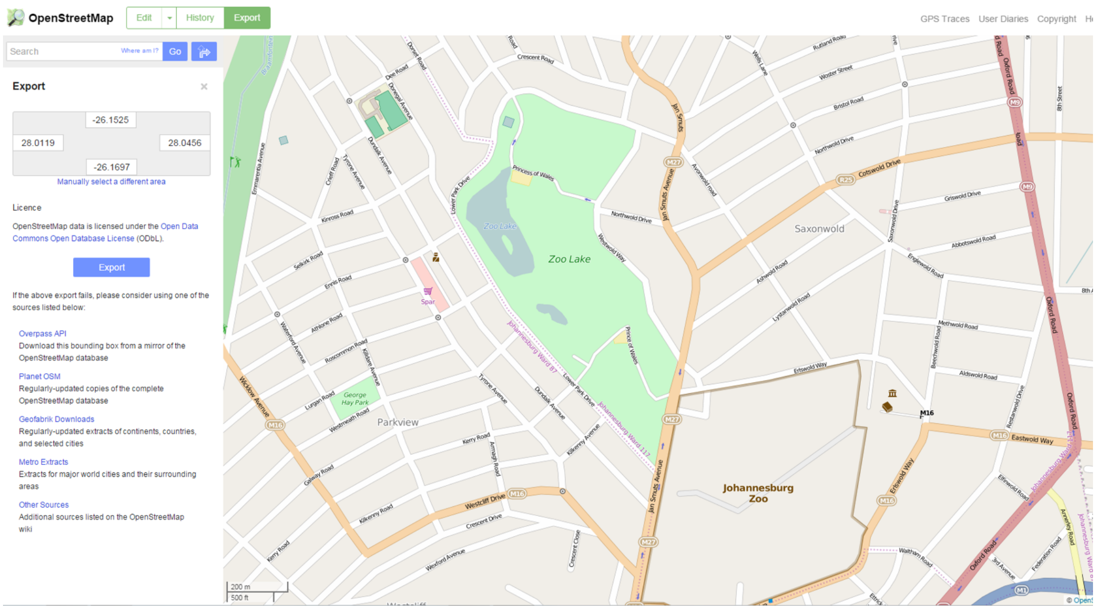


Figure D.1: OSM export window

Table D.1 includes the latitude and longitude settings for the Zoo Lake area used in the OSM export box.

	Minimum	**Maximum**
Latitude	−26,1697°	−26,1525°
Longitude	28,0119°	28,0456°

Table D.1: Zoo Lake settings for OSM export window

Not all the data was required for this study, as only the data referring to roads was needed for the road distances. Therefore, the exported data had to be cleaned to leave only the data referring to roads. In Figure D.2, an example of the exported data from OSM is shown in a text editor.


```xml
</way>
<way id="19045552" visible="true" version="8" changeset="10850751" timestamp="2012-03-02T17:47:15Z" user="dj015" uid="575613">
 <nd ref="197296500"/>
 <nd ref="197307599"/>
 <nd ref="197307644"/>
 <nd ref="197307667"/>
 <nd ref="197307697"/>
 <nd ref="197307721"/>
 <nd ref="197307754"/>
 <nd ref="197307786"/>
 <nd ref="197307812"/>
 <nd ref="197307844"/>
 <tag k="highway" v="residential"/>
 <tag k="name" v="Westwold Way"/>
</way>
<way id="19045654" visible="true" version="4" changeset="17415950" timestamp="2013-08-19T18:55:13Z" user="Rich JL" uid="1713458">
 <nd ref="54994680"/>
 <nd ref="243047449"/>
 <nd ref="197310233"/>
 <tag k="highway" v="residential"/>
 <tag k="name" v="Elfinwold Road"/>
 </node>
 <node id="519643395" lat="-26.1668731" lon="28.0338925"/>
 <node id="519643396" lat="-26.1688775" lon="28.0334534">
   <tag k="highway" v="traffic_signals"/>
 </node>
 <node id="519643397" lat="-26.1658142" lon="28.0341150">
   <tag k="highway" v="traffic_signals"/>
 </node>
 <node id="197296500" lat="-26.1553066" lon="28.0333928"/>
 <node id="197307667" lat="-26.1577701" lon="28.0319619"/>
 <node id="197307844" lat="-26.1623431" lon="28.0348206"/>
 <node id="197295743" lat="-26.1663803" lon="28.0288078">
   <tag k="created by" v="JOSM"/>
```


Figure D.2: An extract from exported OSM data

The exported OSM data was divided into two sections. The first part gave information on the roads ("ways"). A road was identified by a "way id", followed by a set of nodes describing the road ("nd ref"). The second part of the exported data gave information on these nodes. Each node was identified by a "node id", also giving its latitude and longitude. Both the roads and the nodes could also have tags giving additional information.

```
"Westwold Way",30429639,335817049,197297343,197297144
"Pitts",23272491,251851759,452852696
"Kilkenny Road",19045479,197293689,2856756341,197303576,197292868,197305021,197305062,197305095
"Crieff Road",19045487,197304325,197305343,197305442
"Kilkenny Avenue",19045472,197295959,197304520,197304596
"11th Avenue",53549503,59723001,676527285
```

Figure D.3: An extract from cleaned OSM road data

The exported map data was separated into two files, one containing the road data and the other file the node data. Data from both files was cleaned in a text editor, using regular expressions. In Figure D.3, the clean data for six of the 147 roads is shown, while in Figure D.4, the clean data for six of the 4 603 nodes is shown. The cleaned road and node data were saved as CSV files.

```
197296500,-26.1553066,28.0333928
197296550,-26.155787,28.0347289
197296732,-26.1606775,28.0384755
197296783,-26.1623718,28.0397592
197296817,-26.1637254,28.0430604
197296853,-26.1623456,28.0417591
```

Figure D.4: Cleaned node data

For each of the 147 roads on the Zoo Lake metrogaine map, the road nodes describing each road separately were available from the cleaned data files. How these roads interconnect, was needed next. To get the road intersection nodes, a query was run using Overpass turbo, a web-based data filtering tool for OSM (Overpass turbo, 2015). The query that was run is shown in Figure D.5.

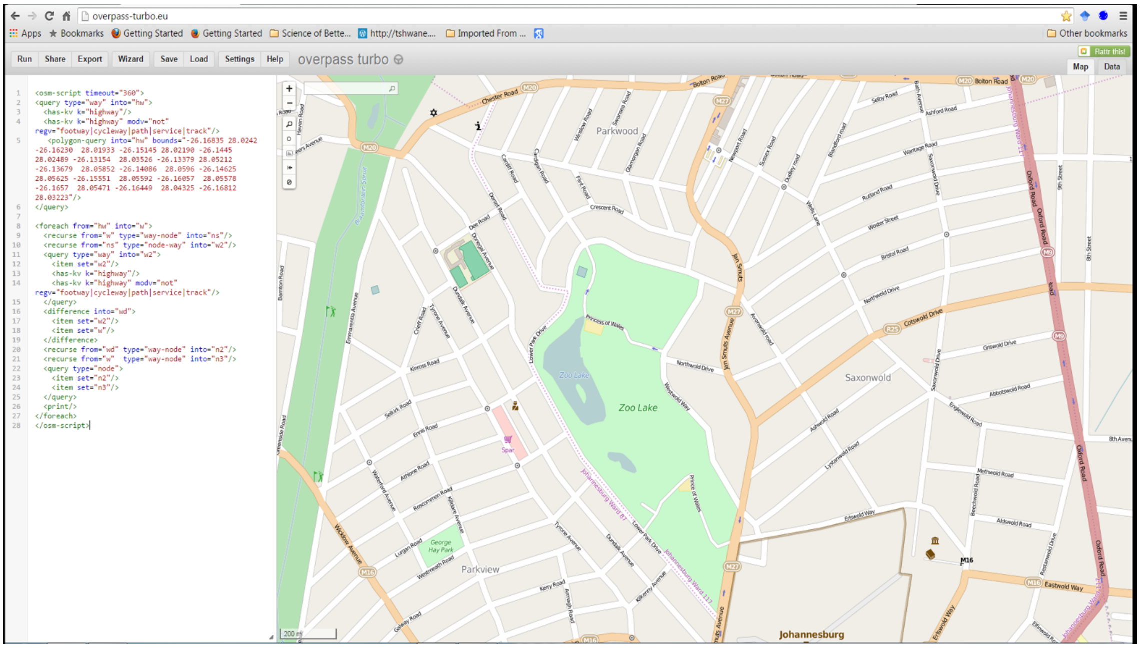


Figure D.5: Overpass turbo query (Overpass turbo, 2015)

A subset of the intersection nodes from the Overpass query on OSM is shown in Figure D.6. Clicking on a node showed the "node id" and its coordinates.

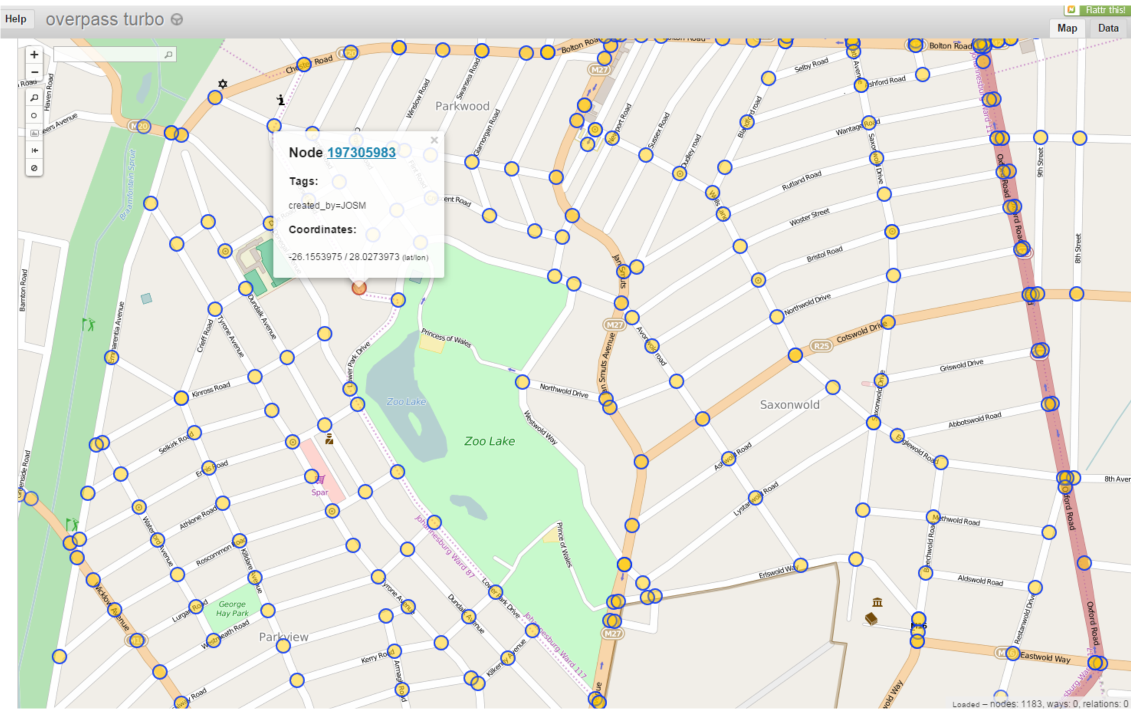


Figure D.6: Intersection nodes on Overpass turbo (Overpass turbo, 2015)

The nodes shown on the left in Figure D.7 are a subset of the intersection nodes from the Overpass query. The intersection node data could be exported in a selection of data formats as shown on the right in Figure D.7. The GPX data format was used in this study.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="Overpass API">
<note>The data included in this document is from www.openstreetmap.org.
<meta osm_base="2015-09-08T09:04:02Z"/>

  <node id="111062680" lat="-26.1818842" lon="28.0214963"/>
  <node id="111065204" lat="-26.1805208" lon="28.0210672">
    <tag k="direction" v="clockwise"/>
    <tag k="highway" v="mini_roundabout"/>
  </node>
  <node id="197292911" lat="-26.1681411" lon="28.0242298">
    <tag k="created_by" v="JOSM"/>
    <tag k="highway" v="traffic_signals"/>
  </node>
  <node id="197293758" lat="-26.1708982" lon="28.0211534"/>
  <node id="197294598" lat="-26.1764092" lon="28.0197662"/>
  <node id="197294619" lat="-26.1761587" lon="28.0198166"/>
  <node id="197295228" lat="-26.1777247" lon="28.0205812">
    <tag k="created_by" v="JOSM"/>
  </node>
  <node id="197295301" lat="-26.1787639" lon="28.0215593">
    <tag k="direction" v="clockwise"/>
    <tag k="highway" v="mini_roundabout"/>
  </node>
  <node id="197295563" lat="-26.1734821" lon="28.0198226"/>
  <node id="197295617" lat="-26.1719660" lon="28.0204806"/>
  <node id="197295672" lat="-26.1697164" lon="28.0223977">
    <tag k="created_by" v="JOSM"/>
  </node>
  <node id="111068304" lat="-26.1831985" lon="28.0314540"/>
  <node id="111072053" lat="-26.1813142" lon="28.0301471"/>
  <node id="197294971" lat="-26.1769922" lon="28.0307079"/>
  <node id="197295890" lat="-26.1687890" lon="28.0333600"/>
  <node id="197295926" lat="-26.1657296" lon="28.0340023">
    <tag k="highway" v="traffic_signals"/>
  </node>
  <node id="197297202" lat="-26.1633006" lon="28.0345202"/>
  <node id="197301316" lat="-26.1747393" lon="28.0320583"/>
  <node id="197301552" lat="-26.1717236" lon="28.0327110"/>
  <node id="197301880" lat="-26.1732646" lon="28.0323690">
    <tag k="highway" v="speed_camera"/>
  </node>
  <node id="197302719" lat="-26.1799517" lon="28.0296928"/>
  <node id="197302831" lat="-26.1764817" lon="28.0312699"/>
  <node id="197304564" lat="-26.1667895" lon="28.0337750"/>
  <node id="197307844" lat="-26.1623431" lon="28.0348206"/>
  <node id="197307919" lat="-26.1637603" lon="28.0344075"/>
  <node id="197309838" lat="-26.1697358" lon="28.0331798">
    <tag k="highway" v="traffic_signals"/>
```

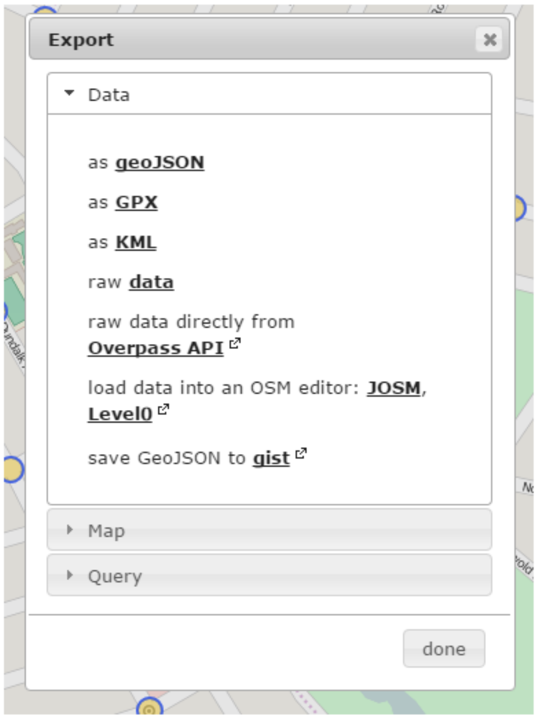


Figure D.7: Intersection nodes from Overpass turbo and available data formats

The intersection data also had to be cleaned. Again the cleaning process was done in a text editor using regular expressions. The cleaned intersection data was saved as a CSV file.

The extraction of the control point attributes was briefly described in Section 3.2 on page 42. No cleaning was required and the attribute data was saved as a CSV file. This attribute CSV file was merged with the cleaned road and node CSV files, as well as the cleaned intersection CSV file, to produce a file containing 4 599 entries. This combined CSV file, named `ZL_map_ data`, described the interconnected roads and control points on the Zoo Lake metrogaine map in a numerical format. Importing `ZL_map_ data` into SAS, the road distances between all the connected nodes were calculated first, followed by the shortest road paths between all the control points.

This shortest path file, `ZL_arcs`, was used as input in all the orienteering problems in this study.

D.2 Visualising the data

Visualisation in uMap was used to verify results (uMap, 2015). Data was imported in GPX format into uMap. Reformatting the CSV data to GPX data was done using a simple Python program.

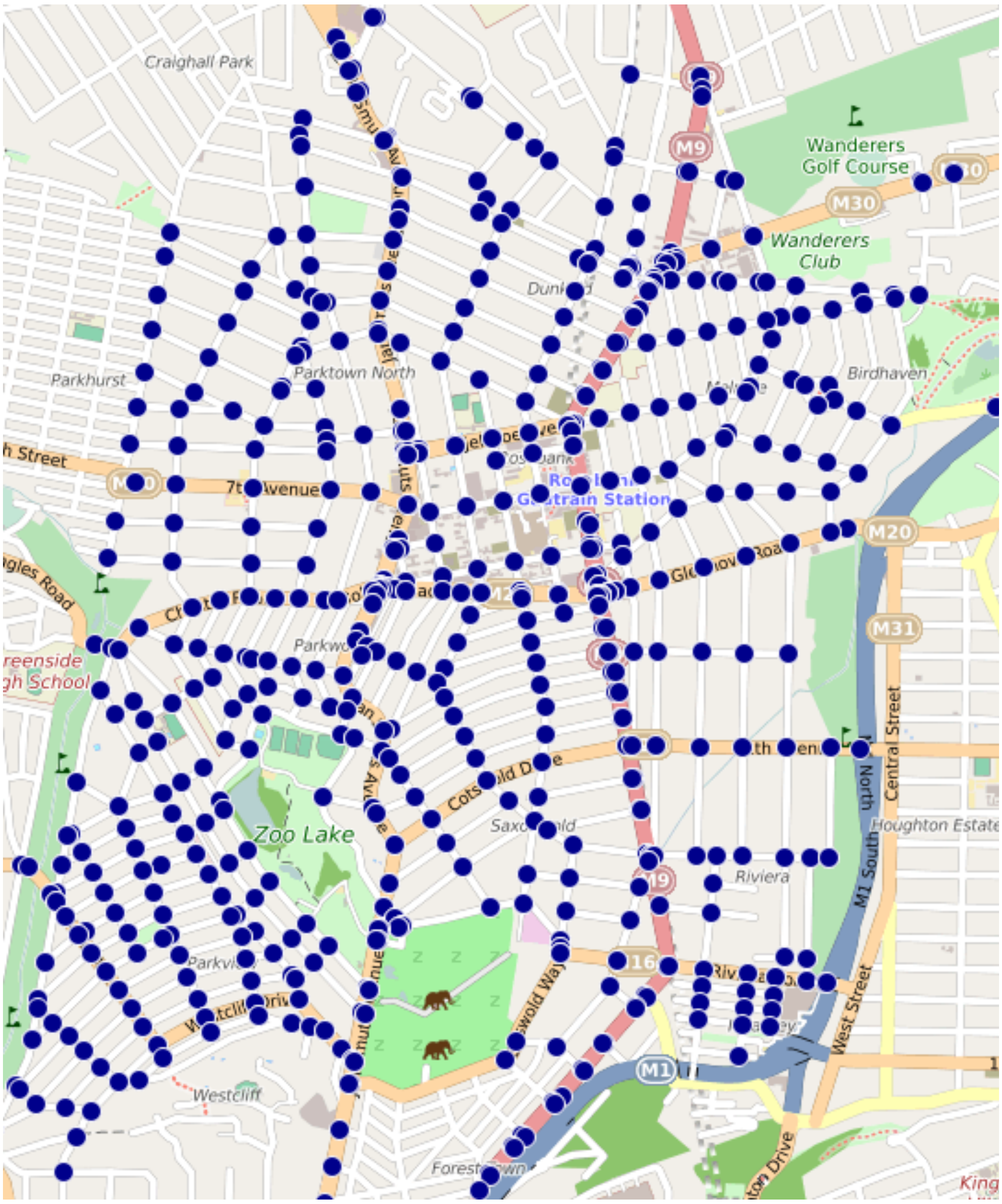


Figure D.8: Intersection nodes from Overpass turbo in uMap

Each file could be displayed as a separate layer on the Zoo Lake map area. Each named layer could be described by its own properties, for example icon shape and colour. The number of layers shown could also be selected. The intersection nodes obtained from the Overpass query is shown in Figure D.8.

In Figure D.9, the location of the control points are shown in uMap.

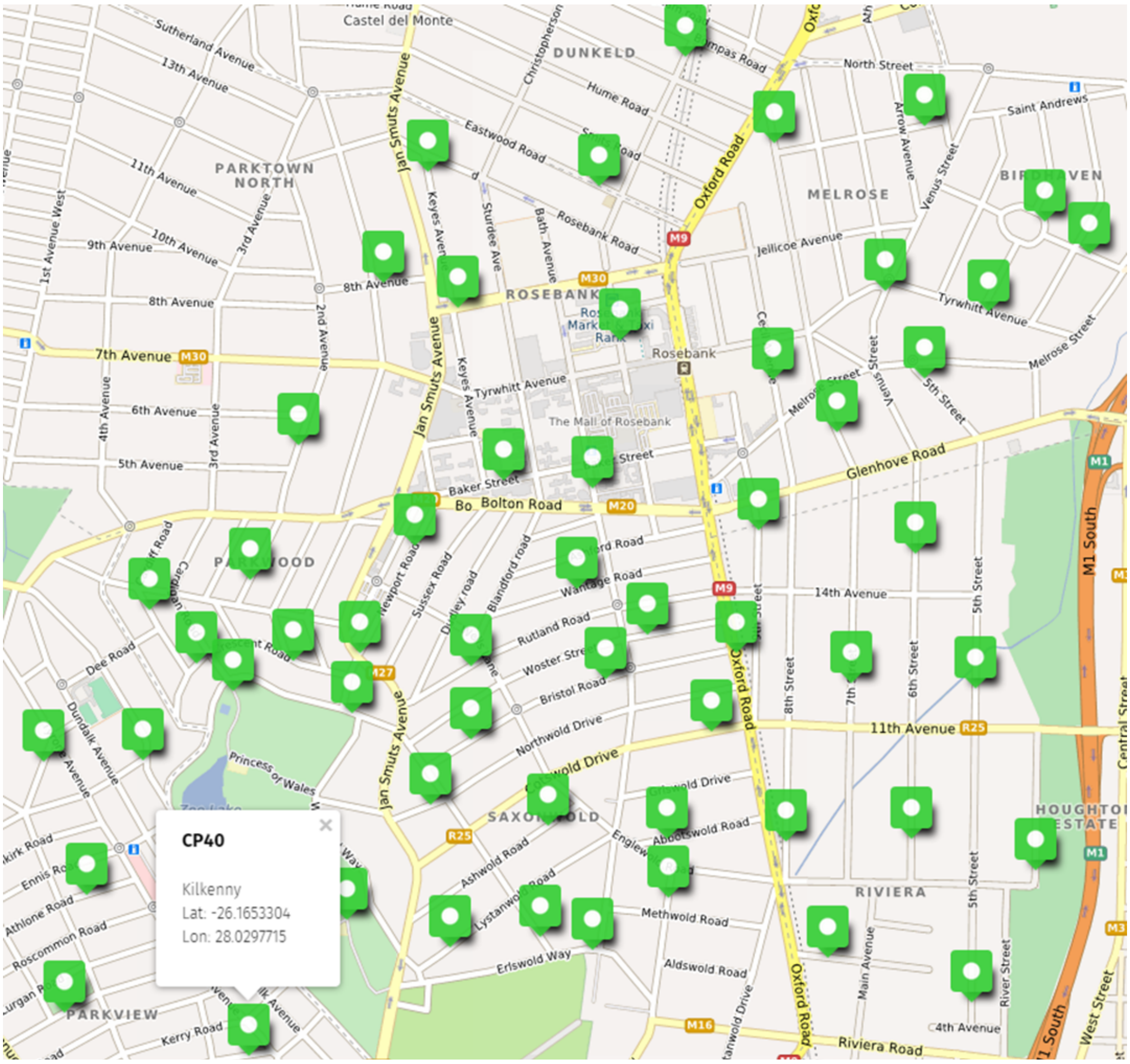


Figure D.9: Control point locations in uMap

Ninety-four shortest road distances between two control points were visually verified in uMap. Five of these paths are shown in Figure D.10.

Figure D.10: Five shortest road paths in uMap

Appendix E

Selected edges

Table E.1: Subset from 1 540 edges

Edge	Geodetic distance (m)	Road distance (m)	Elevation adjusted distance (m)
CP10–CP11	590	949	949
⋮	⋮	⋮	⋮
CP11–CP13	257	334	334
⋮	⋮	⋮	⋮
CP12–CP15	1 420	1 772	1 772
⋮	⋮	⋮	⋮
CP13–CP17	198	226	226
⋮	⋮	⋮	⋮
CP14–CP20	343	512	512
⋮	⋮	⋮	⋮
CP15–CP22	702	826	958
⋮	⋮	⋮	⋮
CP16–CP24	1 685	2 001	2 001
⋮	⋮	⋮	⋮
CP17–CP26	1 615	2 090	2 090
⋮	⋮	⋮	⋮
CP18–CP28	2 060	2 845	3 411

Table E.1 (*continued*)

Edge	**Geodetic distance (m)**	**Road distance (m)**	**Elevation adjusted distance (m)**
CP20–CP30	1 157	1 548	1 548
⋮	⋮	⋮	⋮
CP21–CP32	349	588	588
⋮	⋮	⋮	⋮
CP22–CP34	995	1 108	1 108
⋮	⋮	⋮	⋮
CP23–CP36	910	1 272	1 393
⋮	⋮	⋮	⋮
CP24–CP38	584	662	662
⋮	⋮	⋮	⋮
CP25–CP40	1 180	1 513	1 740
⋮	⋮	⋮	⋮
CP26–CP42	1 502	2 035	2 221
⋮	⋮	⋮	⋮
CP27–CP44	1 564	1 922	1 922
⋮	⋮	⋮	⋮
CP28–CP46	2 700	3 426	3 426
⋮	⋮	⋮	⋮
CP29–CP48	2 326	3 250	3 250
⋮	⋮	⋮	⋮
CP30–CP50	1 896	2 474	2 474
⋮	⋮	⋮	⋮
CP31–CP52	2 003	2 543	2 725
⋮	⋮	⋮	⋮
CP32–CP54	1 157	1 382	1 382
⋮	⋮	⋮	⋮
CP33–CP56	2 137	2 640	2 665
⋮	⋮	⋮	⋮
CP34–CP58	2 496	2 880	3 033

Table E.1 (*continued*)

Edge	**Geodetic distance (m)**	**Road distance (m)**	**Elevation adjusted distance (m)**
CP35–CP60	1 210	1 641	1 840
⋮	⋮	⋮	⋮
CP36–CP62	969	1 346	1 591
⋮	⋮	⋮	⋮
CP37–CP64	591	929	954
⋮	⋮	⋮	⋮
CP38–CP40	2 533	3 313	3 395
⋮	⋮	⋮	⋮
CP39–CP42	660	933	1 106
⋮	⋮	⋮	⋮
CP40–CP44	1 698	2 317	2 317
⋮	⋮	⋮	⋮
CP41–CP46	1 634	2 156	2 156
⋮	⋮	⋮	⋮
CP42–CP48	1 129	1 307	1 307
⋮	⋮	⋮	⋮
CP43–CP50	1 458	1 882	1 894
⋮	⋮	⋮	⋮
CP44–CP52	1 264	1 903	2 387
⋮	⋮	⋮	⋮
CP45–CP54	2 158	2 720	2 720
⋮	⋮	⋮	⋮
CP46–CP56	496	695	935
⋮	⋮	⋮	⋮
CP47–CP58	2 693	3 407	3 421
⋮	⋮	⋮	⋮
CP48–CP60	1 432	2 019	2 243
⋮	⋮	⋮	⋮
CP49–CP61	1 479	1 667	1 667

Table E.1 (*continued*)

Edge	**Geodetic distance (m)**	**Road distance (m)**	**Elevation adjusted distance (m)**
CP50–CP62	1 947	2 399	2 708
⋮	⋮	⋮	⋮
CP51–CP63	2 912	3 311	3 311
⋮	⋮	⋮	⋮
CP52–CP64	1 299	1 833	1 833
⋮	⋮	⋮	⋮
CP53–CP65	1 817	2 426	2 714
⋮	⋮	⋮	⋮
CP54–CP56	1 576	1 928	2 039
⋮	⋮	⋮	⋮
CP55–CP58	2 467	3 213	3 213
⋮	⋮	⋮	⋮
CP56–CP60	1 642	1 877	1 877
⋮	⋮	⋮	⋮
CP57–CP62	1 450	1 743	2 154
⋮	⋮	⋮	⋮
CP58–CP64	564	774	774
⋮	⋮	⋮	⋮
CP59–CP60	2 455	2 963	3 066
⋮	⋮	⋮	⋮
CP60–CP61	482	639	639
⋮	⋮	⋮	⋮
CP61–CP62	2 562	2 988	3 325
⋮	⋮	⋮	⋮
CP62–CP63	1 296	1 718	1 718
⋮	⋮	⋮	⋮
CP63–CP64	507	581	693
⋮	⋮	⋮	⋮
CP64–CP65	774	1 187	1 593

Appendix F

Global and local measures of work

Table F.1: Global work measures

Control point	Geodetic (m)	Road (m)	Elevation adjusted (m)
CP10	372	699	699
CP11	220	323	469
CP12	148	204	204
CP13	395	465	665
CP14	382	489	489
CP15	1 351	1 741	2 301
CP16	2 772	3 388	3 564
CP17	433	596	854
CP18	1 504	1 816	2 346
CP20	363	572	633
CP21	835	1 486	1 633
CP22	785	984	1 413
CP23	1 369	1 553	2 152
CP24	1 121	1 668	2 028
CP25	1 451	1 651	2 191
CP26	2 028	2 502	2 911
CP27	2 506	3 024	3 350
CP28	663	1 103	1 103
CP29	899	1 164	1 283
CP30	823	1 050	1 207

Table F.1 (*continued*)

Control point	**Geodetic (m)**	**Road (m)**	**Elevation adjusted (m)**
CP31	1 186	1 572	1 837
CP32	1 099	1 374	1 710
CP33	746	961	1 280
CP34	1 579	1 951	2 510
CP35	1 654	1 893	2 455
CP36	2 153	2 689	3 168
CP37	2 519	3 238	3 904
CP38	1 448	1 777	2 172
CP39	798	957	1 351
CP40	1 193	1 627	1 939
CP41	1 122	1 383	1 857
CP42	759	988	1 210
CP43	2 136	2 560	3 114
CP44	1 222	1 442	2 010
CP45	3 148	3 867	3 999
CP46	2 043	2 603	3 137
CP47	2 276	2 783	3 201
CP48	1 800	2 161	2 749
CP49	2 283	2 618	2 996
CP50	1 177	1 498	2 040
CP51	1 291	1 450	1 939
CP52	829	1 046	1 130
CP53	1 573	1 899	2 423
CP54	1 880	2 355	2 760
CP55	2 437	2 897	3 166
CP56	2 486	3 018	3 312
CP57	1 705	2 181	2 825
CP58	1 809	2 215	2 620
CP59	2 923	3 642	4 110
CP60	2 697	3 093	3 457
CP61	2 629	3 229	3 799
CP62	3 072	3 757	3 991
CP63	2 545	3 066	3 819
CP64	2 038	2 646	3 287
CP65	1 419	1 680	1 916

Table F.2: Local work measures

Control point	**Geodetic (m)**	**Road (m)**	**Elevation adjusted (m)**
CP10	479	750	761
CP11	319	337	458
CP12	354	467	483
CP13	335	413	424
CP14	435	568	568
CP15	436	532	576
CP16	402	604	613
CP17	322	394	448
CP18	358	575	606
CP20	373	486	547
CP21	504	784	789
CP22	379	511	581
CP23	330	460	496
CP24	451	660	682
CP25	363	527	592
CP26	461	666	682
CP27	433	529	565
CP28	560	739	771
CP29	486	602	643
CP30	498	663	758
CP31	629	790	925
CP32	420	595	636
CP33	394	530	559
CP34	364	534	605
CP35	360	485	510
CP36	395	575	663
CP37	575	748	883
CP38	490	610	649
CP39	347	484	543

Table F.2 (*continued*)

Control point	**Geodetic (m)**	**Road (m)**	**Elevation adjusted (m)**
CP40	603	813	925
CP41	391	547	596
CP42	447	601	621
CP43	518	764	840
CP44	355	460	518
CP45	519	695	695
CP46	449	598	691
CP47	476	668	691
CP48	467	604	647
CP49	505	719	758
CP50	354	514	545
CP51	361	481	540
CP52	588	783	792
CP53	369	589	633
CP54	413	624	624
CP55	569	764	764
CP56	404	663	686
CP57	532	685	807
CP58	634	832	877
CP59	576	736	897
CP60	617	799	818
CP61	676	926	1 094
CP62	450	653	685
CP63	721	893	1 083
CP64	557	745	841
CP65	555	766	797

Appendix G

Tsiligirides distances

Table G.1: The Tsiligirides distances

Control point	**Neighbour distance NN_d (m)**	**Average distance Avg_d (m)**
CP10	522,0	2 388,4
CP11	272,4	1 764,3
CP12	203,5	1 968,2
CP13	225,8	1 651,2
CP14	285,0	2 119,7
CP15	363,9	1 578,1
CP16	376,7	2 461,5
CP17	225,8	1 643,5
CP18	350,9	1 724,3
CP20	386,9	1 893,7
CP21	588,1	2 255,3
CP22	441,1	1 517,2
CP23	269,3	1 521,1
CP24	363,9	1 694,6
CP25	201,6	1 874,9
CP26	603,6	1 857,1
CP27	376,7	2 139,8
CP28	522,0	2 653,4
CP29	457,4	2 588,3
CP30	517,4	2 646,7

Table G.1 (*continued*)

Control point	**Neighbour distance NN_d (m)**	**Average distance Avg_d (m)**
CP31	457,4	2 982,2
CP32	423,2	1 927,7
CP33	356,7	1 696,2
CP34	350,9	1 847,9
CP35	366,0	1 590,3
CP36	462,8	1 915,4
CP37	529,9	2 418,8
CP38	342,7	1 968,7
CP39	356,7	1 591,7
CP40	467,9	2 659,9
CP41	463,1	1 699,7
CP42	392,3	1 648,9
CP43	500,2	2 159,1
CP44	269,3	1 548,0
CP45	209,4	2 832,0
CP46	462,8	1 874,7
CP47	599,6	2 129,1
CP48	509,1	1 659,4
CP49	631,6	1 969,3
CP50	395,2	1 556,2
CP51	201,6	1 869,6
CP52	474,2	2 016,7
CP53	366,0	1 637,4
CP54	500,2	1 882,8
CP55	648,7	2 147,2
CP56	454,1	2 135,9
CP57	570,6	1 817,7
CP58	472,8	2 360,3
CP59	529,9	2 663,9
CP60	599,6	2 491,1
CP61	638,8	2 640,7
CP62	209,4	2 785,6
CP63	548,8	2 534,1
CP64	570,6	2 252,5
CP65	342,7	2 164,4

Appendix H

Fischetti distances

Table H.1: The Fischetti ratios

Control point	**Distance from start CP_d (m)**	**Fischetti ratio**
CP10	699	0,1809
CP11	323	0,0836
CP12	204	0,0526
CP13	465	0,1202
CP14	489	0,1263
CP15	1 741	0,4502
CP16	3 388	0,8761
CP17	596	0,1540
CP18	1 816	0,4697
CP20	572	0,1478
CP21	1 486	0,3844
CP22	984	0,2546
CP23	1 553	0,4015
CP24	1 668	0,4315
CP25	1 651	0,4270
CP26	2 502	0,6471
CP27	3 024	0,7820
CP28	1 103	0,2852
CP29	1 164	0,3009
CP30	1 050	0,2716

Table H.1 (*continued*)

Control point	**Distance from start CP_d (m)**	**Fischetti ratio**
CP31	1 572	0,4064
CP32	1 374	0,3554
CP33	961	0,2484
CP34	1 951	0,5047
CP35	1 893	0,4895
CP36	2 689	0,6954
CP37	3 238	0,8373
CP38	1 777	0,4596
CP39	957	0,2474
CP40	1 627	0,4207
CP41	1 383	0,3577
CP42	988	0,2555
CP43	2 560	0,6619
CP44	1 442	0,3729
CP45	3 867	1,0000
CP46	2 603	0,6732
CP47	2 783	0,7196
CP48	2 161	0,5587
CP49	2 618	0,6769
CP50	1 498	0,3873
CP51	1 450	0,3749
CP52	1 046	0,2704
CP53	1 899	0,4911
CP54	2 355	0,6090
CP55	2 897	0,7493
CP56	3 018	0,7804
CP57	2 181	0,5639
CP58	2 215	0,5728
CP59	3 642	0,9419
CP60	3 093	0,7998
CP61	3 229	0,8351
CP62	3 757	0,9717
CP63	3 066	0,7930
CP64	2 646	0,6843
CP65	1 680	0,4345

Appendix I

Score sets

Table I.1: The `SAME`, `SAME_I`, `SAME_II` and `SAME_III` score sets

Control point	`SAME` **score**	`SAME_I` **score**	`SAME_II` **score**	`SAME_III` **score**
Start/Finish	0	0	0	0
CP10	30	50	60	50
CP11	30	10	10	20
CP12	30	10	30	20
CP13	30	10	10	30
CP14	30	20	20	10
CP15	30	20	30	10
CP16	30	40	30	30
CP17	30	10	30	10
CP18	30	20	30	10
CP20	30	10	10	40
CP21	30	50	50	60
CP22	30	30	20	60
CP23	30	10	10	20
CP24	30	20	10	20
CP25	30	30	30	40
CP26	30	50	60	50
CP27	30	40	30	30
CP28	30	50	30	60
CP29	30	50	30	40
CP30	30	50	30	20

Table I.1 (*continued*)

Control point	SAME score	SAME_I score	SAME_II score	SAME_III score
CP31	30	50	50	40
CP32	30	30	30	50
CP33	30	10	30	20
CP34	30	20	10	10
CP35	30	20	30	10
CP36	30	40	30	40
CP37	30	40	40	30
CP38	30	20	10	10
CP39	30	10	30	10
CP40	30	50	50	30
CP41	30	30	20	50
CP42	30	40	30	60
CP43	30	50	30	40
CP44	30	10	30	20
CP45	30	40	30	30
CP46	30	40	40	30
CP47	30	50	30	40
CP48	30	30	30	50
CP49	30	60	60	50
CP50	30	10	10	20
CP51	30	20	20	20
CP52	30	20	30	20
CP53	30	30	20	50
CP54	30	40	50	50
CP55	30	60	30	50
CP56	30	40	50	40
CP57	30	60	30	60
CP58	30	40	40	50
CP59	30	30	30	30
CP60	30	60	60	40
CP61	30	60	60	40
CP62	30	30	40	30
CP63	30	30	30	30
CP64	30	30	40	10
CP65	30	20	30	10

Table I.2: The EXIST, RANDOM, TSP and FISCHETTI score sets

Control point	EXIST **score**	RANDOM **score**	TSP **score**	FISCHETTI **score**
Start/Finish	0	0	0	0
CP10	10	20	10	10
CP11	10	20	10	10
CP12	10	40	10	10
CP13	10	30	10	10
CP14	10	40	10	10
CP15	10	10	50	30
CP16	10	10	40	60
CP17	10	20	10	10
CP18	10	10	30	30
CP20	20	50	10	10
CP21	20	20	20	20
CP22	20	20	40	20
CP23	20	10	50	30
CP24	20	20	40	20
CP25	20	10	30	30
CP26	20	10	50	40
CP27	20	40	40	50
CP28	20	20	10	10
CP29	20	50	20	20
CP30	30	30	10	20
CP31	30	20	20	30
CP32	30	10	20	20
CP33	30	40	40	10
CP34	30	40	30	40
CP35	30	40	50	40
CP36	30	50	40	50
CP37	30	40	30	50
CP38	30	50	20	30
CP39	30	50	40	20
CP40	40	60	20	30

Table I.2 (*continued*)

Control point	EXIST **score**	RANDOM **score**	TSP **score**	FISCHETTI **score**
CP41	40	10	30	20
CP42	40	50	40	10
CP43	40	30	60	50
CP44	40	40	50	30
CP45	40	40	30	60
CP46	40	40	40	40
CP47	40	40	60	50
CP48	40	30	50	40
CP49	40	30	50	50
CP50	50	60	50	20
CP51	50	40	30	30
CP52	50	10	10	20
CP53	50	30	60	40
CP54	50	50	60	40
CP55	50	40	50	50
CP56	50	40	40	50
CP57	50	40	20	40
CP58	50	50	20	40
CP59	50	50	30	60
CP60	60	20	50	60
CP61	60	40	60	50
CP62	60	50	30	60
CP63	60	60	30	50
CP64	60	20	20	40
CP65	60	30	20	30

Table I.3: The `TSILIG100SI0SII`, `TSILIG50SI50SII` and `TSILIG0SI100SII` score sets

Control point	`TSILIG100SI0SII` **score**	`TSILIG50SI50SII` **score**	`TSILIG0SI100SII` **score**
Start/Finish	0	0	0
CP10	50	50	50
CP11	10	20	20
CP12	10	20	30
CP13	10	10	20
CP14	20	30	40
CP15	20	10	10
CP16	30	50	50
CP17	10	10	10
CP18	20	20	20
CP20	30	30	30
CP21	50	50	40
CP22	30	10	10
CP23	10	10	10
CP24	20	20	20
CP25	10	20	30
CP26	60	30	20
CP27	30	40	40
CP28	50	60	50
CP29	30	50	50
CP30	50	60	50
CP31	40	60	60
CP32	30	30	30
CP33	20	20	20
CP34	20	30	20
CP35	20	10	10
CP36	40	30	30
CP37	50	50	50
CP38	20	30	30
CP39	20	10	10
CP40	40	50	60

Table I.3 (*continued*)

Control point	`TSILIG100SI0SII` **score**	`TSILIG50SI50SII` **score**	`TSILIG0SI100SII` **score**
CP41	40	20	20
CP42	30	20	10
CP43	40	40	40
CP44	10	10	10
CP45	10	50	60
CP46	40	30	30
CP47	50	40	40
CP48	40	20	20
CP49	60	40	30
CP50	30	10	10
CP51	10	20	30
CP52	40	40	40
CP53	30	10	10
CP54	40	30	30
CP55	60	40	40
CP56	30	40	40
CP57	50	30	20
CP58	40	40	50
CP59	50	60	60
CP60	60	50	50
CP61	60	60	50
CP62	10	50	60
CP63	50	50	50
CP64	50	40	40
CP65	20	40	40

Table I.4: The Geo100G0L, Geo80G20L, Geo50G50L, Geo20G80L and Geo0G100L score sets

Control point	Geo100G0L **score**	Geo80G20L **score**	Geo50G50L **score**	Geo20G80L **score**	Geo0G100L **score**
Start/Finish	0	0	0	0	0
CP10	10	10	10	10	40
CP11	10	10	10	10	10
CP12	10	10	10	10	10
CP13	10	10	10	10	10
CP14	10	10	10	10	30
CP15	30	30	30	30	30
CP16	60	60	50	50	20
CP17	10	10	10	10	10
CP18	30	30	30	30	10
CP20	10	10	10	10	20
CP21	20	20	20	20	40
CP22	20	20	10	10	20
CP23	30	30	30	20	10
CP24	20	20	20	30	30
CP25	30	30	30	30	20
CP26	40	40	40	40	40
CP27	50	50	50	50	30
CP28	10	10	20	30	50
CP29	20	20	20	20	40
CP30	20	20	20	20	40
CP31	30	30	30	40	60
CP32	20	20	20	20	30
CP33	10	10	10	10	20
CP34	40	40	40	30	20
CP35	40	40	40	30	10
CP36	50	50	40	40	20
CP37	50	50	50	50	50
CP38	30	30	30	30	40

Table I.4 (*continued*)

Control point	`Geo100G0L` **score**	`Geo80G20L` **score**	`Geo50G50L` **score**	`Geo20G80L` **score**	`Geo0G100L` **score**
CP39	20	20	10	10	10
CP40	30	30	30	40	50
CP41	20	20	20	20	20
CP42	10	10	20	20	30
CP43	50	50	50	50	40
CP44	30	30	30	20	10
CP45	60	60	60	60	50
CP46	40	40	40	40	30
CP47	50	50	50	50	40
CP48	40	40	40	40	40
CP49	50	50	50	50	40
CP50	20	20	20	20	10
CP51	30	30	30	20	20
CP52	20	20	20	30	50
CP53	40	40	30	30	20
CP54	40	40	40	40	30
CP55	50	50	50	50	50
CP56	50	50	50	40	30
CP57	40	40	40	40	50
CP58	40	40	40	50	60
CP59	60	60	60	60	50
CP60	60	60	60	60	60
CP61	50	50	60	60	60
CP62	60	60	60	50	30
CP63	50	50	50	60	60
CP64	40	40	50	50	50
CP65	30	30	40	40	50

Table I.5: The `Road100G0L`, `Road80G20L`, `Road50G50L`, `Road20G80L` and `Road0G100L` score sets

Control point	`Road100G0L` **score**	`Road80G20L` **score**	`Road50G50L` **score**	`Road20G80L` **score**	`Road0G100L` **score**
Start/Finish	0	0	0	0	0
CP10	10	10	10	20	50
CP11	10	10	10	10	10
CP12	10	10	10	10	10
CP13	10	10	10	10	10
CP14	10	10	10	10	20
CP15	30	30	30	30	20
CP16	60	60	60	50	30
CP17	10	10	10	10	10
CP18	30	30	30	30	20
CP20	10	10	10	10	10
CP21	20	30	30	40	50
CP22	10	10	10	10	20
CP23	30	30	30	20	10
CP24	30	30	30	30	40
CP25	30	30	30	20	20
CP26	40	40	40	40	40
CP27	50	50	50	40	20
CP28	20	20	20	30	50
CP29	20	20	20	20	30
CP30	20	20	20	20	40
CP31	30	30	30	40	50
CP32	20	20	20	20	30
CP33	10	10	10	10	20
CP34	40	40	40	30	20
CP35	40	40	30	30	10
CP36	50	50	40	40	30
CP37	60	50	50	50	50
CP38	30	30	30	30	30

Table I.5 (*continued*)

Control point	Road100G0L score	Road80G20L score	Road50G50L score	Road20G80L score	Road0G100L score
CP39	10	10	10	10	10
CP40	30	30	30	40	60
CP41	20	20	20	20	20
CP42	20	20	20	20	30
CP43	40	40	50	50	50
CP44	20	20	20	10	10
CP45	60	60	60	60	40
CP46	40	40	40	40	30
CP47	50	50	50	50	40
CP48	40	40	40	30	30
CP49	50	50	50	50	40
CP50	30	20	20	20	20
CP51	20	20	20	20	10
CP52	20	20	20	30	50
CP53	40	40	40	30	30
CP54	40	40	40	40	30
CP55	50	50	50	50	50
CP56	50	50	50	50	40
CP57	40	40	40	40	40
CP58	40	40	40	50	60
CP59	60	60	60	60	40
CP60	50	50	50	50	60
CP61	50	60	60	60	60
CP62	60	60	60	60	40
CP63	50	50	50	60	60
CP64	50	50	50	50	50
CP65	30	30	40	40	50

Table I.6: The Alt_adj100G0L, Alt_ad 80G20L, Alt_adj50G50L, Alt_adj20G80L and Alt_adj0G100L score sets

Control point	Alt_adj100G0L score	Alt_adj80G20L score	Alt_adj50G50L score	Alt_adj20G80L score	Alt_adj0G100L score
Start/Finish	0	0	0	0	0
CP10	10	10	10	10	40
CP11	10	10	10	10	10
CP12	10	10	10	10	10
CP13	10	10	10	10	10
CP14	10	10	10	10	20
CP15	30	30	30	30	20
CP16	50	50	50	50	30
CP17	10	10	10	10	10
CP18	30	30	30	30	20
CP20	10	10	10	10	20
CP21	20	20	20	30	50
CP22	20	20	20	20	20
CP23	30	30	30	20	10
CP24	30	30	30	30	30
CP25	30	30	30	30	20
CP26	40	40	40	40	40
CP27	50	50	50	40	20
CP28	10	10	10	20	50
CP29	20	20	20	20	30
CP30	10	20	20	20	40
CP31	20	20	30	40	60
CP32	20	20	20	20	30
CP33	20	20	10	10	20
CP34	40	40	40	30	20
CP35	40	40	40	30	10
CP36	50	50	50	40	30
CP37	60	60	60	60	50

Table I.6 (*continued*)

Control point	Alt_adj100G0L score	Alt_adj80G20L score	Alt_adj50G50L score	Alt_adj20G80L score	Alt_adj0G100L score
CP38	30	30	30	30	30
CP39	20	20	20	10	10
CP40	30	30	30	40	60
CP41	20	20	20	20	20
CP42	20	10	10	10	30
CP43	40	50	50	50	50
CP44	30	30	20	20	10
CP45	60	60	60	60	40
CP46	50	40	40	40	40
CP47	50	50	50	40	40
CP48	40	40	40	40	30
CP49	40	40	40	50	40
CP50	30	30	30	20	10
CP51	30	20	20	20	10
CP52	10	10	20	30	50
CP53	40	40	40	30	30
CP54	40	40	40	40	30
CP55	50	50	50	50	40
CP56	50	50	50	50	40
CP57	40	40	40	50	50
CP58	40	40	40	50	50
CP59	60	60	60	60	60
CP60	50	50	50	50	50
CP61	50	50	60	60	60
CP62	60	60	50	50	40
CP63	60	60	60	60	60
CP64	50	50	50	50	50
CP65	20	30	30	40	50

Appendix J

SAS code

J.1 Shortest path code

Documented SAS code: SAS (2015)

```
/* Shortest paths between all control points */

proc optnet
graph_direction = undirected
data_nodes_sub  = data.NodeSetInSub_AllCP
data_links      = data.LinkSetInZL;
    data_links_var
       from      = node1
       to        = node2
       weight    = distance_m;
    shortpath
       out_weights = data.ShortPathW_ZL_AllCP
       out_paths   = data.ShortPathP_ZL_AllCP;
run;
```

J.2 OP code: row generation model

OrienteeringRowGen4.sas: Pratt (2015a)

```
%let distance_budget = 16000;

%macro findConnectedComponents;
   if card(EDGES_SOL) > 0 then do;
      solve with NETWORK /
      graph_direction = undirected
         links    = (include=EDGES_SOL)
         subgraph = (nodes=NODES_SOL)
         concomp
         out      = (concomp=component_id);
      COMPONENT_IDS = setof {i in NODES_SOL} component_id[i];
      for {c in COMPONENT_IDS} COMPONENT[c] = {};
      for {i in NODES_SOL} do;
         ci = component_id[i];
         COMPONENT[ci] = COMPONENT[ci] union {i};
      end;
   end;
   else COMPONENT_IDS = {};
%mend findConnectedComponents;


proc optmodel printlevel=0;
   /* declare parameters and read data */
   set NODES;
   num x {NODES};
   num y {NODES};
   num score {NODES};
   str name {NODES};
   read data data.zl_same_scores into NODES=[control_point]
x=longitude y=latitude score;
```

```
   num source;
    for {i in NODES: score[i] = 0} do;
      source = i;
      leave;
   end;
   set <num,num> EDGES;
   num distance {EDGES};
   read data data.zl_arcs into EDGES=[source sink]
distance=path_weight;

   /* declare optimation model */
   var UseNode {NODES} binary;
   var UseEdge {EDGES} binary;
   max TotalScore = sum {i in NODES} score[i] * UseNode[i];
   con TwoMatching {i in NODES}:
      sum {<(i),j> in EDGES} UseEdge[i,j]
    + sum {<j,(i)> in EDGES} UseEdge[j,i]
    = 2 * UseNode[i];
   con DistanceCon:
      sum {<i,j> in EDGES} distance[i,j] * UseEdge[i,j]
<= &distance_budget;
   fix UseNode[source] = 1;

   num num_subtours init 0;
   /* subset of nodes not containing source node */
   set SUBTOUR {1..num_subtours};

   /* if node k in SUBTOUR[s] is used, then must use at least
two edges across partition induced by SUBTOUR[s] */
   con SubtourElimination
      {s in 1..num_subtours, k in SUBTOUR[s]}:
      sum {i in NODES diff SUBTOUR[s], j in
SUBTOUR[s]: <i,j> in EDGES} UseEdge[i,j]
    + sum {i in SUBTOUR[s], j in NODES diff
SUBTOUR[s]: <i,j> in EDGES} UseEdge[i,j]
    >= 2 * UseNode[k];
```

```
num num_components;
set NODES_SOL = {i     in NODES: UseNode[i].sol   > 0.5};
set EDGES_SOL = {<i,j> in EDGES: UseEdge[i,j].sol > 0.5};
num component_id {NODES_SOL};
set COMPONENT_IDS;
set COMPONENT {COMPONENT_IDS};
num ci;
num target init constant('BIG');

/* row generation loop */
do until (num_components <= 1);
   /* call MILP solver */
   put 'Solving MILP master...';
   solve with MILP / target=(target);
   target = min(target, _OROPTMODEL_NUM_['BEST_BOUND']);

   /* create data set for use by PROC SGPLOT */
   create data data.sganno_16same from [i j]=EDGES_SOL
      drawspace='datavalue' function='line'
       x1=x[i] y1=y[i] x2=x[j] y2=y[j];

  /* plot current solution */
  submit score=_OBJ_.sol distance=DistanceCon.body
   NumNodesVisited=(card(NODES_SOL));
  title "Total score = &score; Scores:Same;
   Distance = &distance; &NumNodesVisited CPs";
  proc sgplot data=data.zl_same_scores aspect=1
   sganno=data.sganno_16same;
   scatter x=longitude y=latitude
   / datalabel=control_point legendlabel="CP number";
  scatter x=longitude y=latitude / datalabel=score
  run;
  endsubmit;

 /* create output data set */
```

```
    create data data.score_16same from TotalScore
     distance=DistanceCon.body NumNodesVisited=(card(NODES_SOL));
    create data data.sortedCP_16same from [node]=NODES_SOL;

    /* check connectivity of solution */
    %findConnectedComponents;
    num_components = card(COMPONENT_IDS);

    /* create subtour from each component not containing source node */
    for {k in COMPONENT_IDS: source not in COMPONENT[k]} do;
    num_subtours = num_subtours + 1;
    SUBTOUR[num_subtours] = COMPONENT[k];
    put SUBTOUR[num_subtours]=;
    end;
 end;
set <num,num,num> ID_ORDER_NODE;
solve with network / cycle links=(include=EDGES_SOL)
     out=(cycles=ID_ORDER_NODE);
create data data.routeCP_16same from [id order node]=ID_ORDER_NODE;
quit;
```

J.3 OP code: flow-based model

OrienteeringFlow.sas: Pratt (2015a)

```
%let node_data = data.zl_same_i_scores;
%let edge_data = data.zl_arcs;
%let distance_budget = 10000;

title 'Locations and Scores';
proc sgplot data=&node_data aspect=1;
scatter x=longitude y=latitude / datalabel=control_point;
   scatter x=longitude y=latitude / datalabel=score;
run;
title;

proc optmodel printlevel=0;
   /* declare parameters and read data */
   set NODES;
   num x {NODES};
   num y {NODES};
   num score {NODES};
   str name {NODES};
   read data &node_data into NODES=[control_point] x=longitude
y=latitude score;
   num source;
   for {i in NODES: score[i] = 0} do;
      source = i;
      leave;
   end;
   set <num,num> EDGES;
   set ARCS init EDGES;
   num distance {ARCS};
   read data &edge_data into EDGES=[source sink] distance=path_weight;
   for {<i,j> in EDGES} do;
```

```
      ARCS = ARCS union {<j,i>};
      distance[j,i] = distance[i,j];
   end;

   /* declare optimization model */
   var UseNode {NODES} binary;
   var UseArc  {ARCS}  binary;
   max TotalScore = sum {i in NODES} score[i] * UseNode[i];
   con LeaveNode {i in NODES}:
      sum {<(i),j> in ARCS} UseArc[i,j] = UseNode[i];
   con EnterNode {i in NODES}:
      sum {<j,(i)> in ARCS} UseArc[j,i] = UseNode[i];
   con DistanceCon:
      sum {<i,j> in ARCS} distance[i,j] * UseArc[i,j]
<= &distance_budget;
   fix UseNode[source] = 1;

   /* enforce connectivity with flow-based model */
   set COMMODITIES = NODES diff {source};
   var Flow {ARCS, COMMODITIES} >= 0;

   /* if UseNode[k] = 1 then send one unit of commodity k
    from source to node k */
   con Balance {i in NODES, k in COMMODITIES}:
      sum {<(i),j> in ARCS} Flow[i,j,k]
    - sum {<j,(i)> in ARCS} Flow[j,i,k]
    = (if i = source then 1 else if i = k then -1 else 0) * UseNode[k];
   /* if Flow[i,j,k] > 0 then UseArc[i,j] = 1 */
   con Link {<i,j> in ARCS, k in COMMODITIES}:
      Flow[i,j,k] <= UseArc[i,j];

   /* if nodes i and j are too far from source, cannot use both nodes */
   set CONFLICTS = {<i,j> in EDGES: source not in {i,j} and
      distance[source,i] + distance[i,j] + distance[j,source]
> &distance_budget};
   con Cut {<i,j> in CONFLICTS}:
```

```
      UseNode[i] + UseNode[j] <= 1;

   set NODES_SOL = {i     in NODES: UseNode[i].sol  > 0.5};
   set ARCS_SOL  = {<i,j> in ARCS:  UseArc[i,j].sol > 0.5};

   /* call MILP solver */
   solve;

   /* create data set for use by PROC SGPLOT */
   create data data.sganno_10same_i from [i j]=ARCS_SOL
      drawspace='datavalue' function='line'
      x1=x[i] y1=y[i] x2=x[j] y2=y[j];

   /* plot current solution */
   submit score=_OBJ_.sol distance=DistanceCon.body
NumNodesVisited=(card(NODES_SOL));
      title "Total score = &score; Scores:same_i; Distance = &distance;
&NumNodesVisited CPs";
proc sgplot data=data.zl_same_i_scores aspect=1
sganno=data.sganno_10same_i;
    scatter x=longitude y=latitude / datalabel=control_point
legendlabel="CP number";
scatter x=longitude y=latitude / datalabel=score legendlabel="Score";
      run;
   endsubmit;

/* create output data set */
create data data.score_10same_i from TotalScore
distance=DistanceCon.body NumNodesVisited=(card(NODES_SOL));
create data data.sortedCP_10same_i from [node]=NODES_SOL;
set <num,num,num> ID_ORDER_NODE;
solve with network / cycle links=(include=ARCS_SOL)
out=(cycles=ID_ORDER_NODE);
create data data.routeCP_10same_i from [id order node]=ID_ORDER_NODE;
quit;
```

J.4 OP code: relaxation and repair

OrienteeringRepair.sas: Pratt (2017)

```
%let distance_budget = 9000;
%let relobjgap       = 0.01;

title 'Control point locations';
proc sgplot data=data.ZL_FISCH_scores aspect=1;
   scatter x=longitude y=latitude / datalabel=control_point;
run;
title;

%macro findConnectedComponents;
   if card(EDGES_SOL) > 0 then do;
      put 'Finding connected components...';
      solve with network / concomp
         links=(include=EDGES_SOL) subgraph=(nodes=NODES_SOL)
         out=(concomp=component);
      COMPONENTS = setof {i in NODES_SOL} component[i];
      for {c in COMPONENTS} NODES_c[c] = {};
      for {i in NODES_SOL} do;
         ci = component[i];
         NODES_c[ci] = NODES_c[ci] union {i};
      end;
   end;
   else COMPONENTS = {};
   num_components= card(COMPONENTS);
   put num_components=;
   /* create subtour from each component not containing depot node */
   for {c in COMPONENTS: depot not in NODES_c[c]} do;
      num_subtours = num_subtours + 1;
      SUBTOUR[num_subtours] = NODES_c[c];
      put SUBTOUR[num_subtours]=;
```

```
   end;
%mend findConnectedComponents;

%macro checkFeasibility;
   if (DistanceCon.body <= DistanceCon.ub) then do;
      lower_bound = _OBJ_.sol;
      if best_lower_bound < lower_bound then do;
         best_lower_bound = lower_bound;
      end;
   end;
%mend checkFeasibility;

%macro repairSolution;
   put 'Solving TSP...';
   solve with network / tsp
      links=(weight=distance) subgraph=(nodes=NODES_SOL)
      out=(tour=TOUR);
   for {<i,j> in EDGES} UseEdge[i,j] = (<i,j> in TOUR);
%mend repairSolution;

proc optmodel printlevel=0;
   /* declare parameters and read data */
   set NODES;
   num x {NODES};
   num y {NODES};
   num score {NODES};
   str name {NODES};
   read data data.ZL_FISCH_scores into NODES=[control_point]
    x=longitude y=latitude score;
   num depot;
   for {i in NODES: score[i] = 0} do;
      depot = i;
      leave;
   end;
   set <num,num> EDGES;
   num distance {EDGES};
```

```
   read data data.zl_arcs into EDGES=[source sink]
    distance=path_weight;

   /* declare optimation model */
   var UseNode {NODES} binary;
   var UseEdge {EDGES} binary;
   max TotalScore = sum {i in NODES} score[i] * UseNode[i];
   con TwoMatching {i in NODES}:
      sum {<(i),j> in EDGES} UseEdge[i,j]
    + sum {<j,(i)> in EDGES} UseEdge[j,i]
    = 2 * UseNode[i];
   con DistanceCon:
      sum {<i,j> in EDGES} distance[i,j] * UseEdge[i,j]
        <= &distance_budget;
   con UseDepot:
      UseNode[depot] = 1;

   num num_subtours init 0;
   /* subset of nodes not containing depot node */
   set SUBTOUR {1..num_subtours};

   /* if node k in SUBTOUR[s] is used, then  must use at least
    two edges across partition induced by SUBTOUR[s] */
*   con SubtourElimination
      {s in 1..num_subtours, k in SUBTOUR[s]}:
      sum {i in NODES diff SUBTOUR[s], j in SUBTOUR[s]:
       <i,j> in EDGES}  UseEdge[i,j]
    + sum {i in SUBTOUR[s], j in NODES diff SUBTOUR[s]:
     <i,j> in EDGES}
     UseEdge[i,j]  >= 2 * UseNode[k];
   /* sparser version using TwoMatching constraints */
   con SubtourElimination {s in 1..num_subtours, k in SUBTOUR[s]}:
      sum {i in SUBTOUR[s], j in SUBTOUR[s]: <i,j> in EDGES}
       UseEdge[i,j] <= sum {i in SUBTOUR[s] diff {k}} UseNode[i];

   num epsilon = 1e-3;
```

```
num num_components;
set NODES_SOL = {i     in NODES: UseNode[i].sol   > epsilon};
set EDGES_SOL = {<i,j> in EDGES: UseEdge[i,j].sol > epsilon};
num component {NODES_SOL};
set COMPONENTS;
set NODES_c {COMPONENTS};
num ci;

set <num,num> TOUR;

num infinity = constant('BIG');
num lower_bound, upper_bound;
num best_lower_bound init 0;
num best_upper_bound init infinity;
num gap = (if best_lower_bound ne 0 and best_upper_bound < infinity
          then (best_upper_bound - best_lower_bound) /
           abs(best_upper_bound)
          else .);

/* row generation loop */
do until ((gap ne . and gap <= &relobjgap) or
 _solution_status_ = 'INFEASIBLE');
   do until (num_components <= 1);
      /* call LP solver */
      put 'Solving LP master...';
      solve with LP relaxint;
      upper_bound = _OROPTMODEL_NUM_['OBJECTIVE'];
      best_upper_bound = min(best_upper_bound, upper_bound);
      put "BOUNDS: " best_lower_bound= best_upper_bound= gap=;
      /* plot LP solution */
      create data data.sganno_9FISCH from [i j]=EDGES_SOL
         drawspace='datavalue' function='line'
          linethickness=UseEdge[i,j]
         x1=x[i] y1=y[i] x2=x[j] y2=y[j];
      submit score=_OBJ_.sol distance=DistanceCon.body
       NumNodesVisited=(card(NODES_SOL))
```

```
    lb=best_lower_bound ub=best_upper_bound;
      title "Total score = &score; Scores:Fischetti;
       Distance = &distance; &NumNodesVisited CPs";
      title2 "Bounds: [&lb, &ub]";
      proc sgplot data=data.ZL_FISCH_scores aspect=1
       sganno=data.sganno_9FISCH;
        scatter x=longitude y=latitude / datalabel=control_point
         legendlabel="CP number";
        scatter x=longitude y=latitude / datalabel=score
         legendlabel="Score";
      run;
   endsubmit;
   %findConnectedComponents;
end;
/* call MILP solver */
put 'Solving MILP master...';
solve with MILP / relobjgap=&relobjgap target=(best_upper_bound);
upper_bound = _OROPTMODEL_NUM_['BEST_BOUND'];
best_upper_bound = min(best_upper_bound, upper_bound);
put "BOUNDS: " best_lower_bound= best_upper_bound= gap=;
%findConnectedComponents;
/* plot MILP solution */
create data data.sganno_9FISCH from [i j]=EDGES_SOL
   drawspace='datavalue' function='line'
   x1=x[i] y1=y[i] x2=x[j] y2=y[j];
submit score=_OBJ_.sol distance=DistanceCon.body
 NumNodesVisited=(card(NODES_SOL))
 lb=best_lower_bound ub=best_upper_bound;
   title "Total score = &score; Scores:Fischetti;
    Distance = &distance; &NumNodesVisited CPs";
 proc sgplot data=data.ZL_FISCH_scores aspect=1
  sganno=data.sganno_9FISCH;
   scatter x=longitude y=latitude / datalabel=control_point
    legendlabel="CP number";
   scatter x=longitude y=latitude / datalabel=score
    legendlabel="Score";
```

```
         run;
      endsubmit;
      %repairSolution;
      %checkFeasibility;
      put "BOUNDS: " best_lower_bound= best_upper_bound= gap=;
      /* plot TSP solution */
      create data data.sganno_9FISCH from [i j]=EDGES_SOL
         drawspace='datavalue' function='line'
         x1=x[i] y1=y[i] x2=x[j] y2=y[j];
      submit score=_OBJ_.sol distance=DistanceCon.body
       NumNodesVisited=(card(NODES_SOL))
       lb=best_lower_bound ub=best_upper_bound;
         title "Total score = &score; Scores:Fischetti;
          Distance = &distance; &NumNodesVisited CPs";
      proc sgplot data=data.ZL_FISCH_scores aspect=1
       sganno=data.sganno_9FISCH;
         scatter x=longitude y=latitude / datalabel=control_point
          legendlabel="CP number";
         scatter x=longitude y=latitude / datalabel=score
          legendlabel="Score";
         run;
      endsubmit;
   end;

   /* create output data sets */
   set <num,num,num> ID_ORDER_NODE;
   solve with network / cycle links=(include=EDGES_SOL)
     out=(cycles=ID_ORDER_NODE);
   create data data.routeCP_9FISCH from [id order node]=ID_ORDER_NODE;
   create data data.score_9FISCH from TotalScore
    distance=DistanceCon.body NumNodesVisited=(card(NODES_SOL));
   create data data.sortedCP_9FISCH from [node]=NODES_SOL;
quit;
```

Appendix K

Graphical OP output

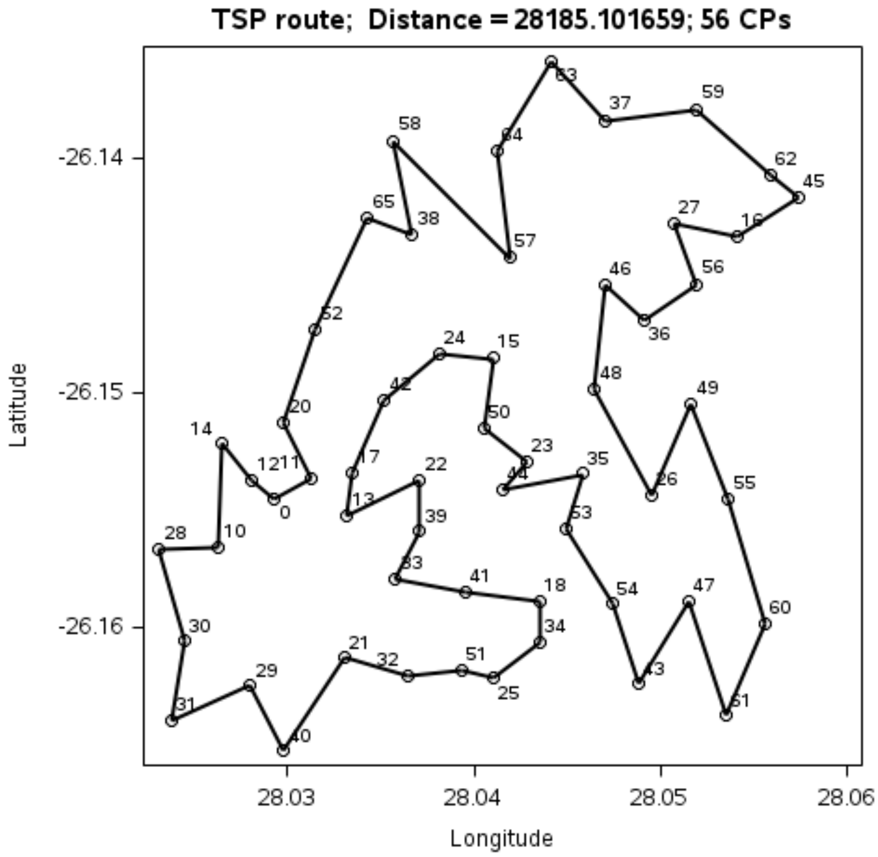
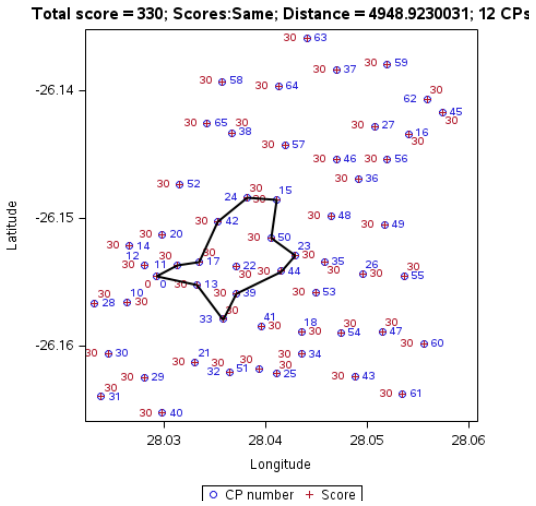
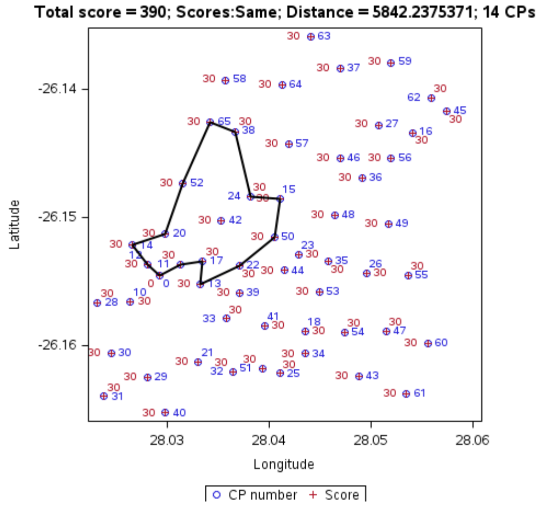


Figure K.1: The TSP route of 28,2 km for all score sets

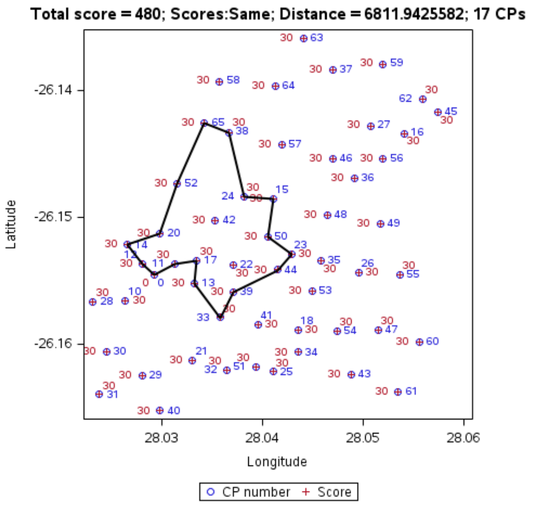
K.1 SAME OP output



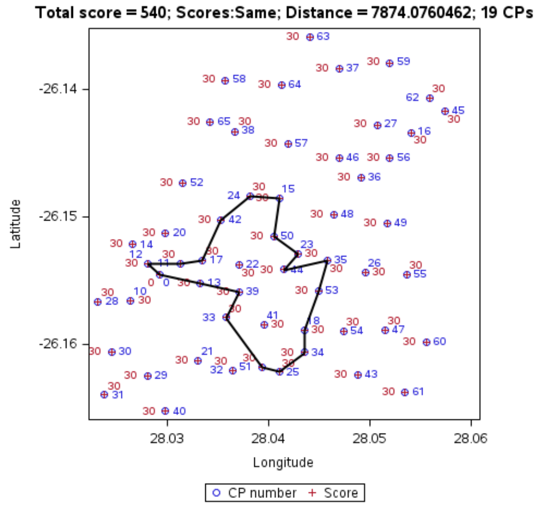
(a) 5 km



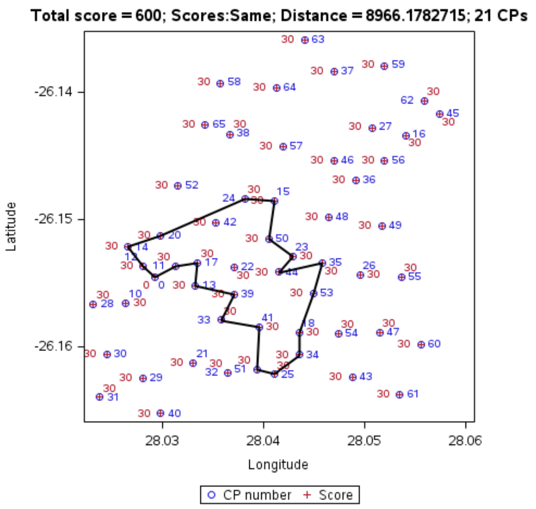
(b) 6 km



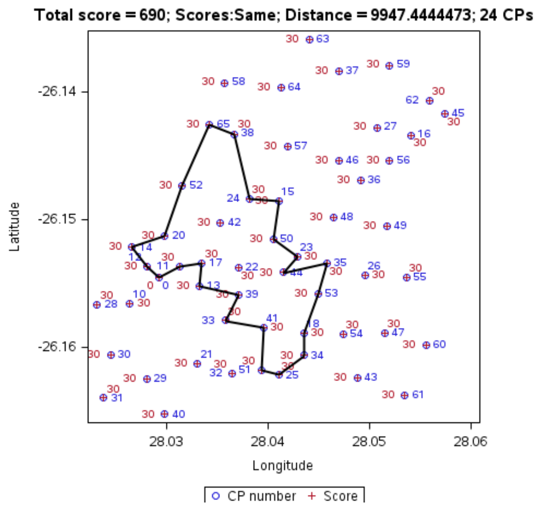
(c) 7 km



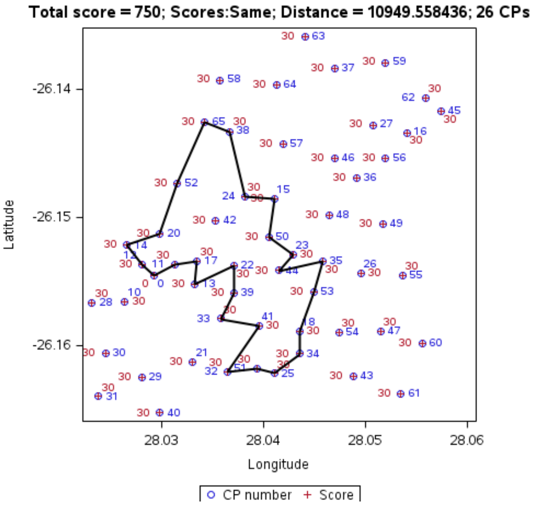
(d) 8 km



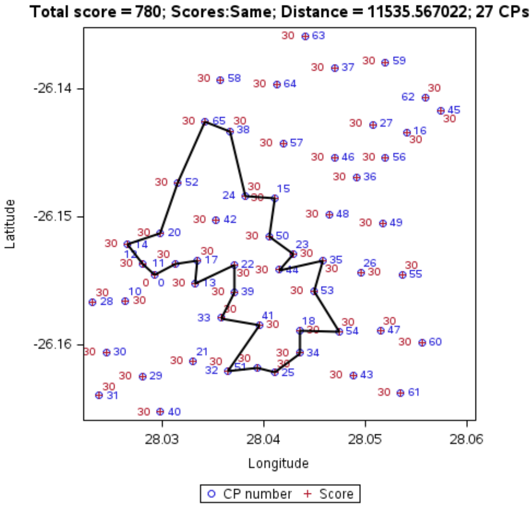
(e) 9 km



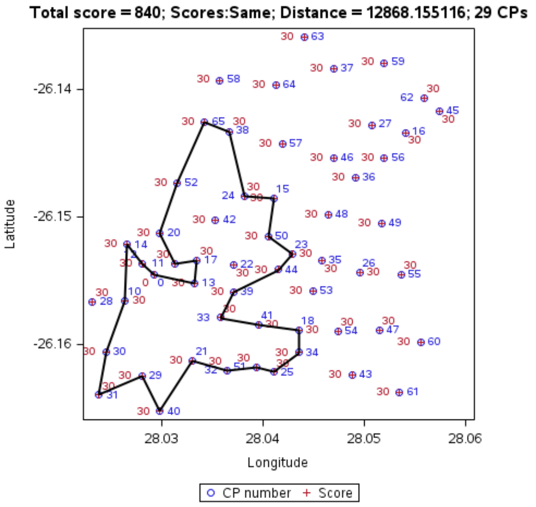
(f) 10 km



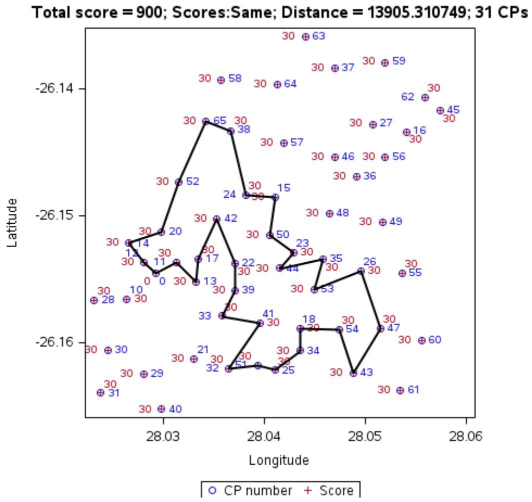
(g) 11 km



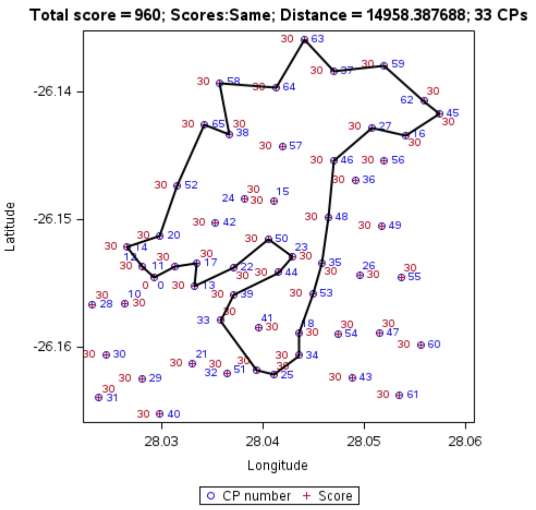
(h) 12 km



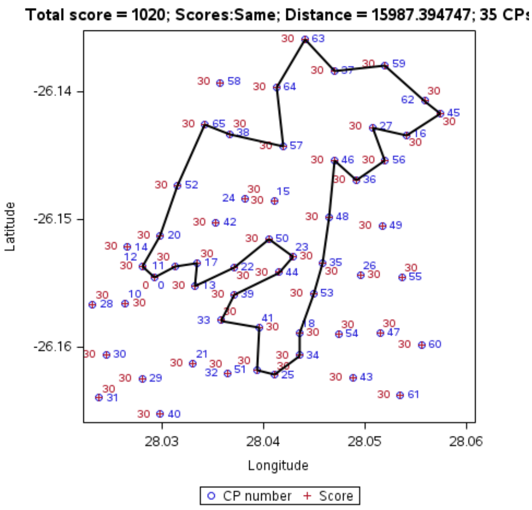
(i) 13 km



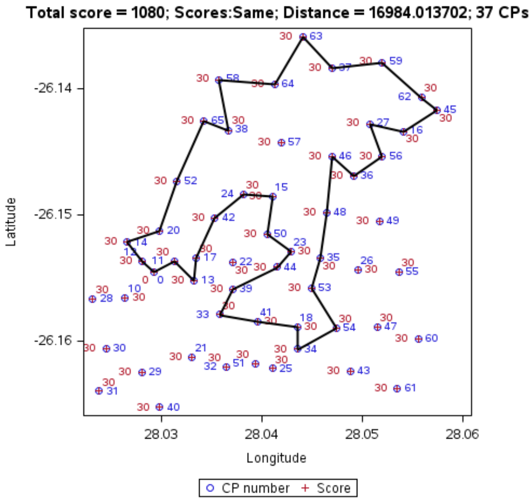
(j) 14 km



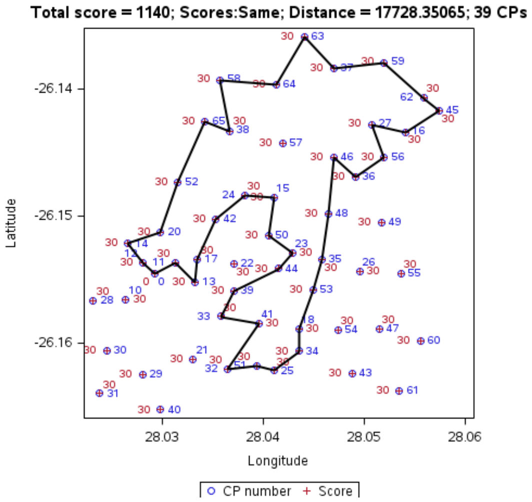
(k) 15 km



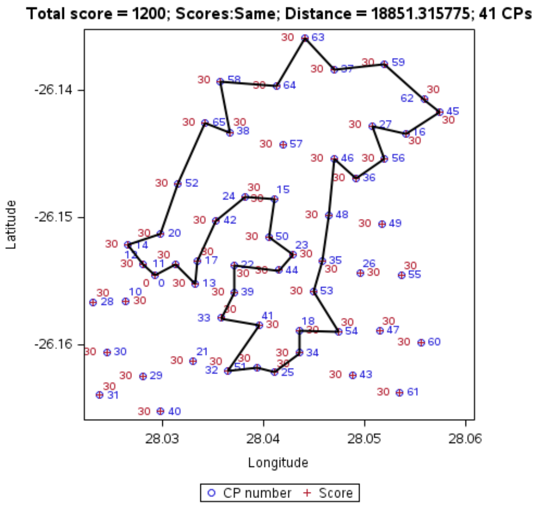
(l) 16 km



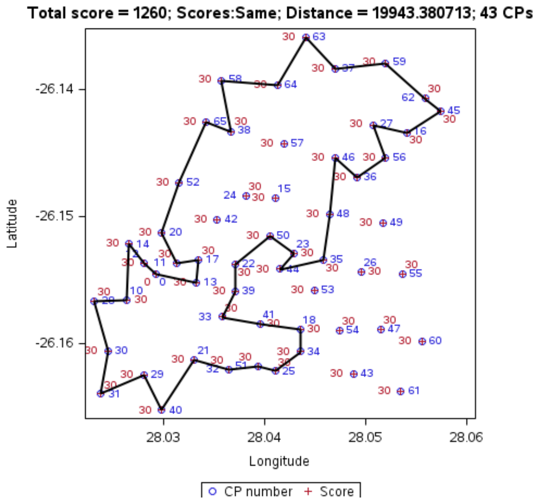
(m) 17 km



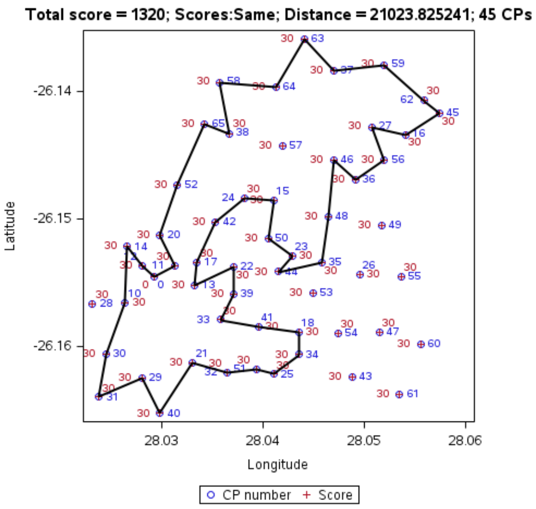
(n) 18 km



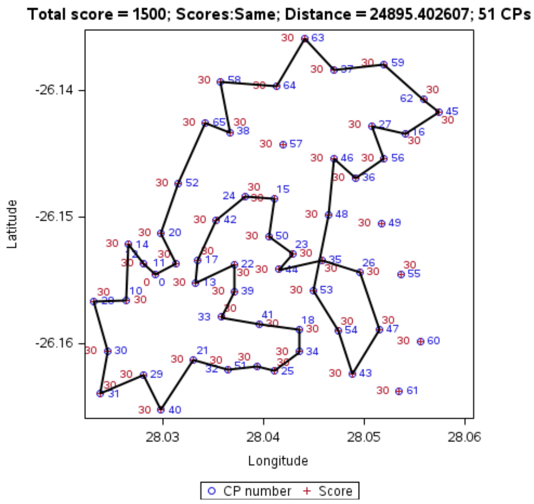
(o) 19 km



(p) 20 km

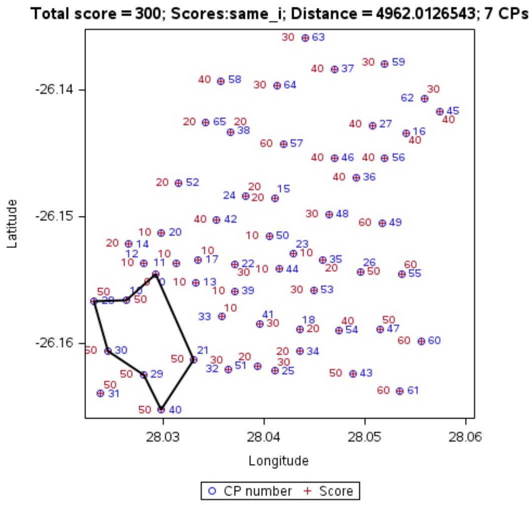


(q) 21,1 km

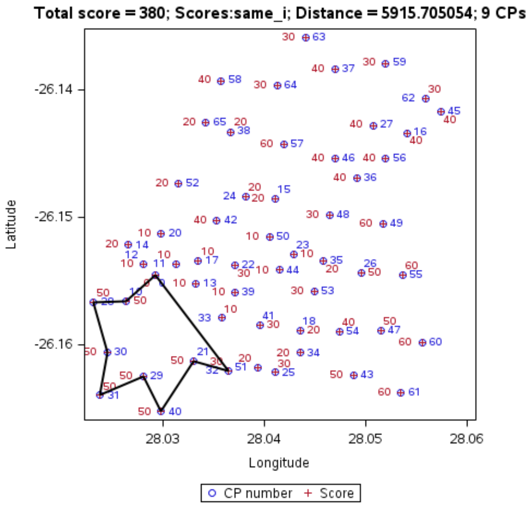


(r) 25 km

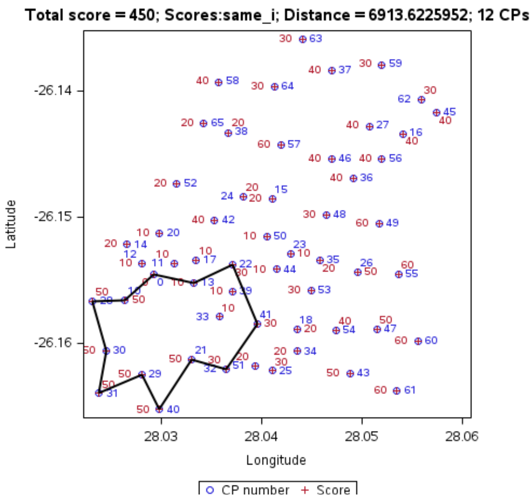
K.2 SAME_I OP output



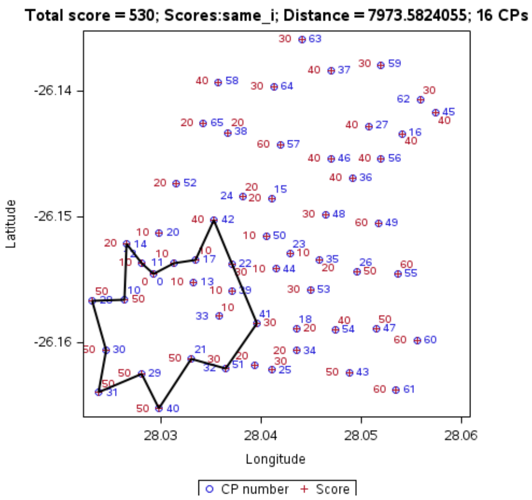
(a) 5 km



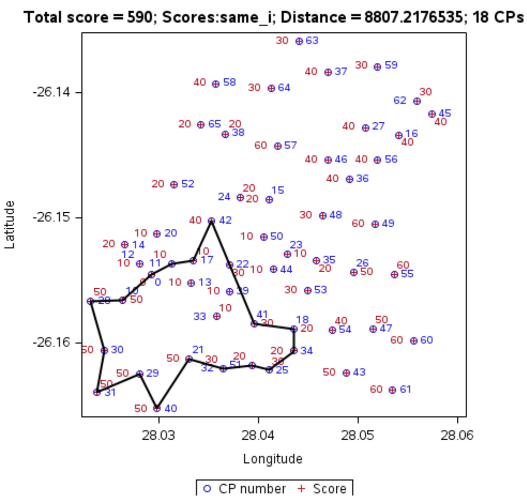
(b) 6 km



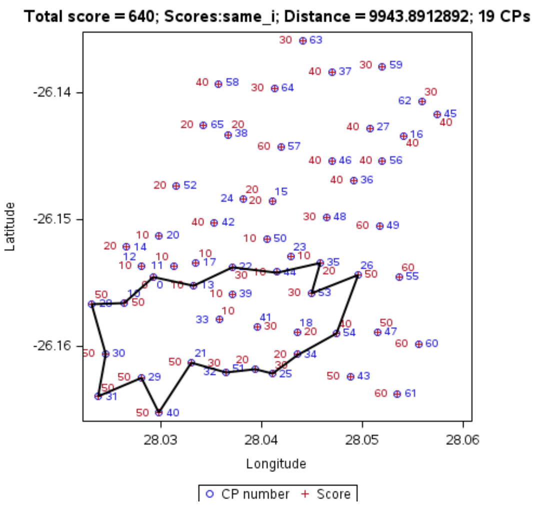
(c) 7 km



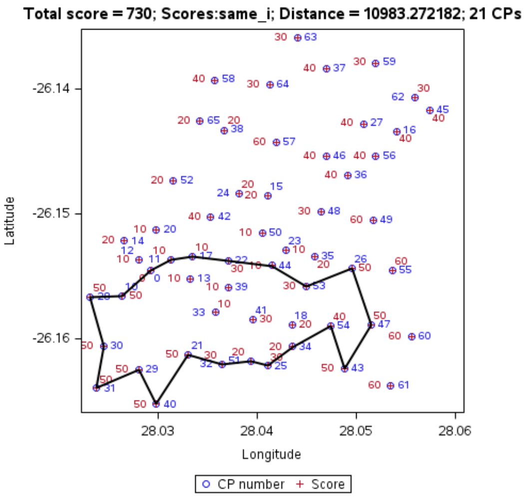
(d) 8 km



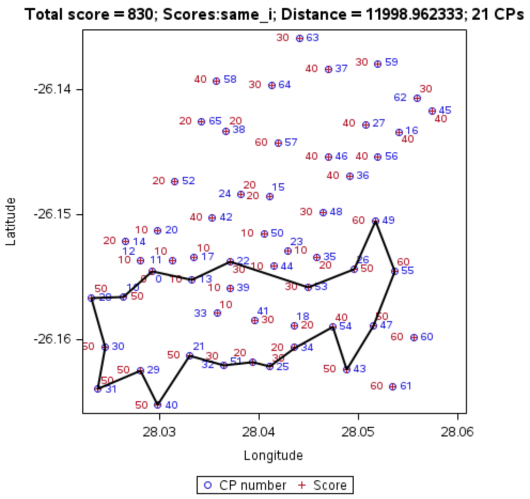
(e) 9 km



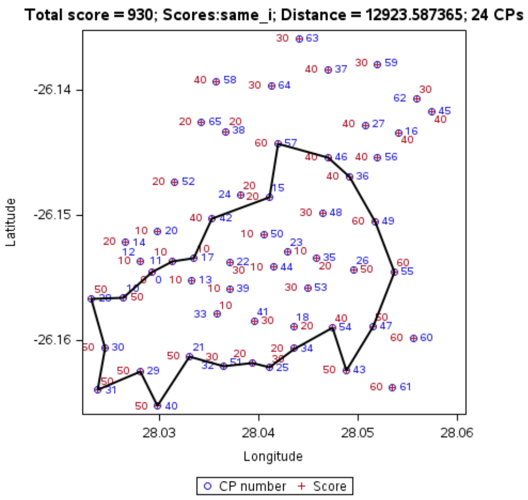
(f) 10 km



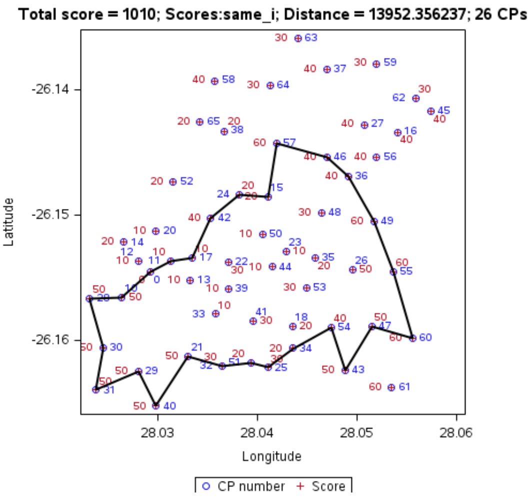
(g) 11 km



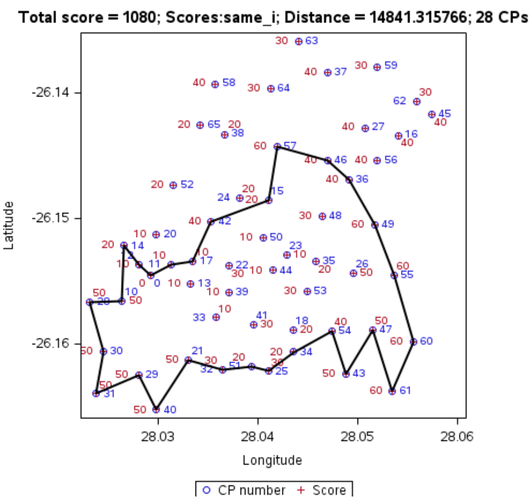
(h) 12 km



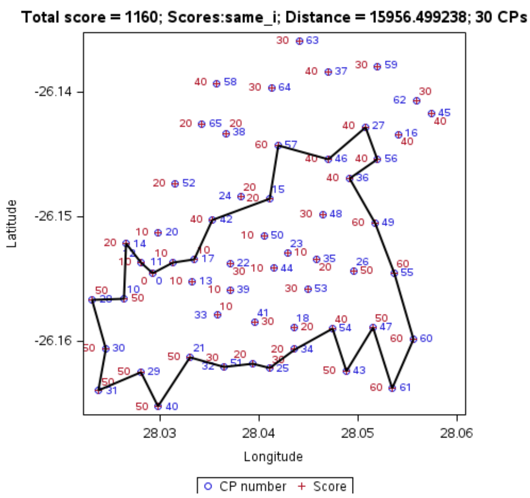
(i) 13 km



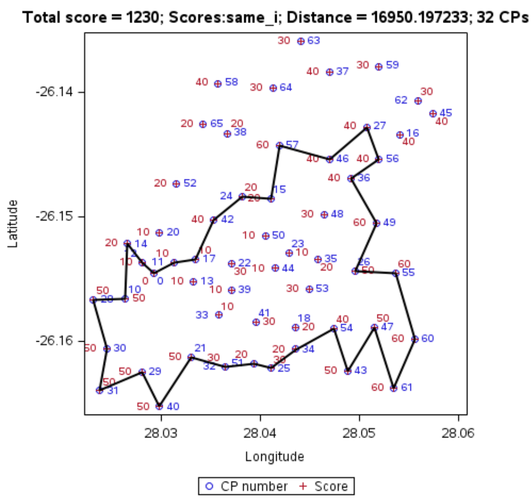
(j) 14 km



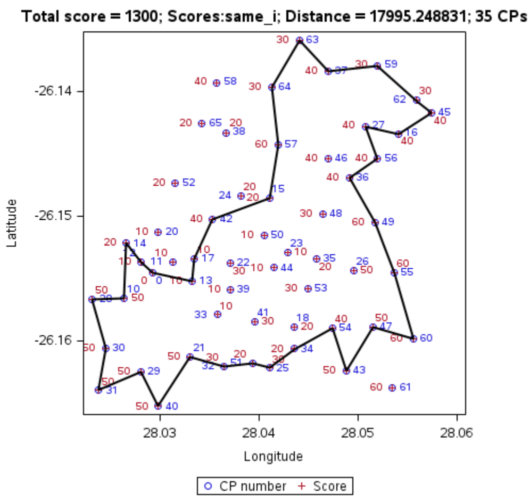
(k) 15 km



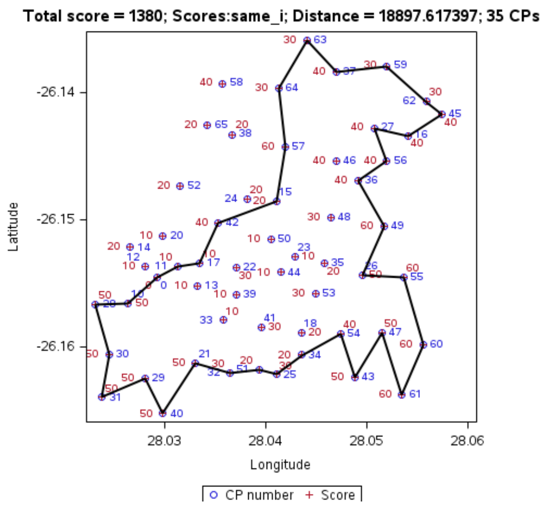
(l) 16 km



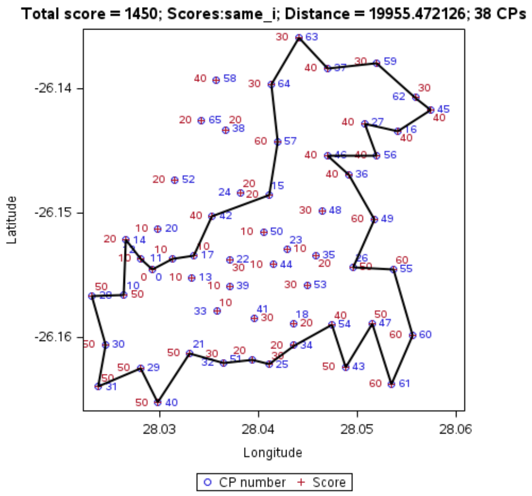
(m) 17 km



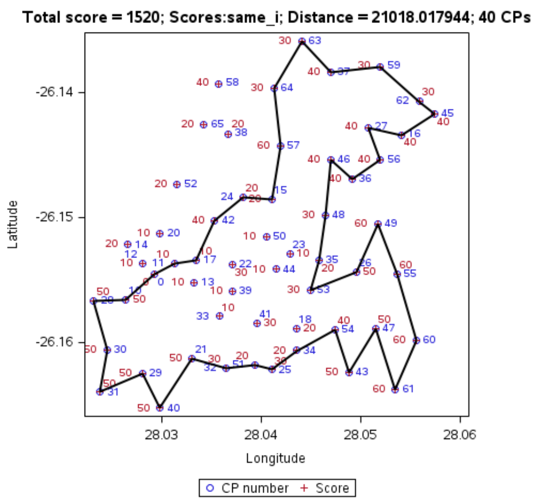
(n) 18 km



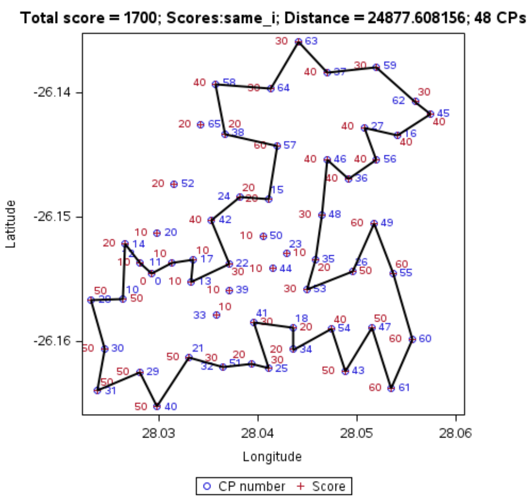
(o) 19 km



(p) 20 km

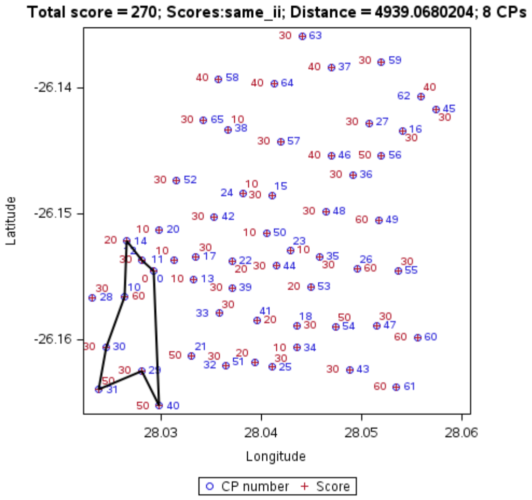


(q) 21,1 km

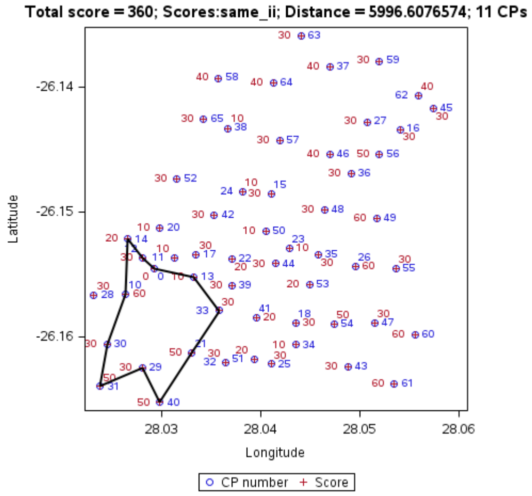


(r) 25 km

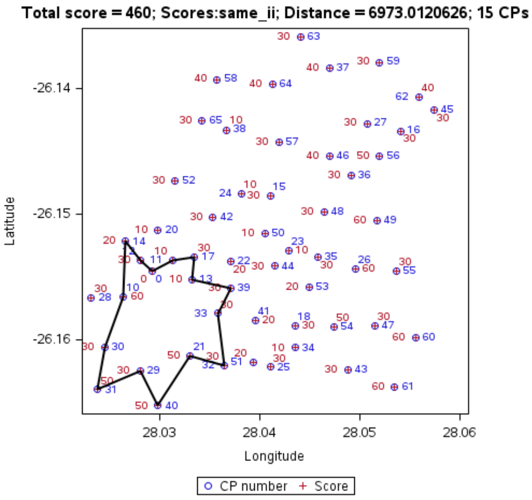
K.3 SAME_II OP output



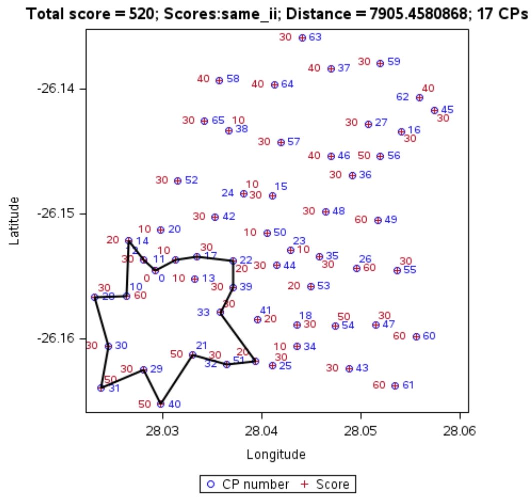
(a) 5 km



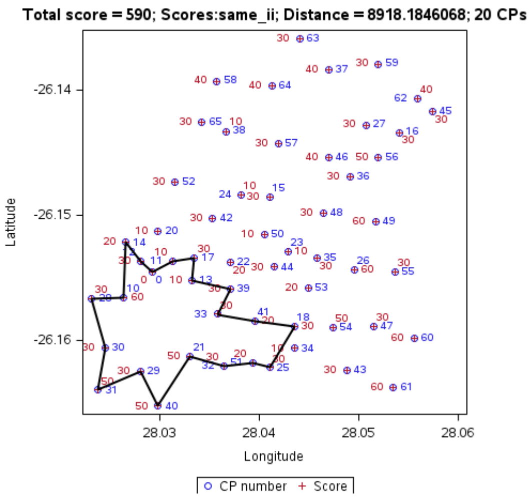
(b) 6 km



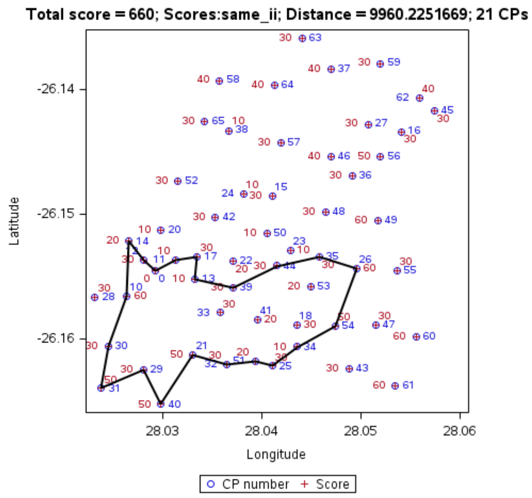
(c) 7 km



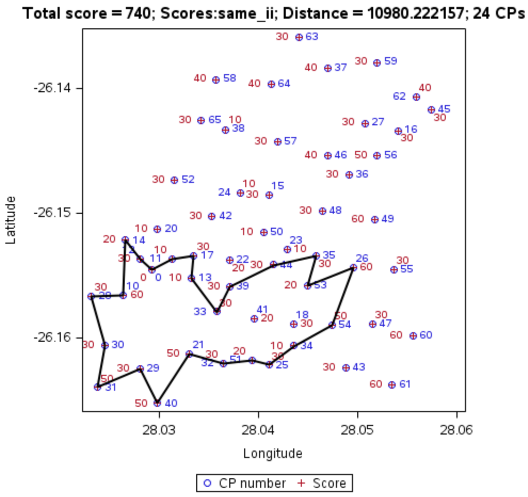
(d) 8 km



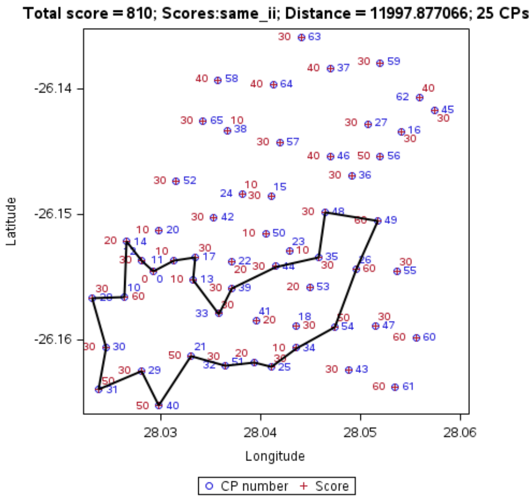
(e) 9 km



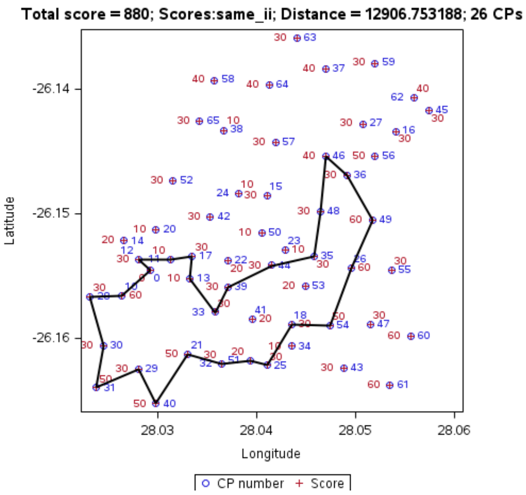
(f) 10 km



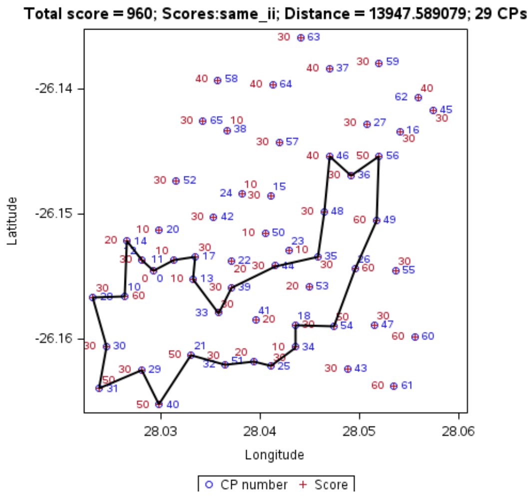
(g) 11 km



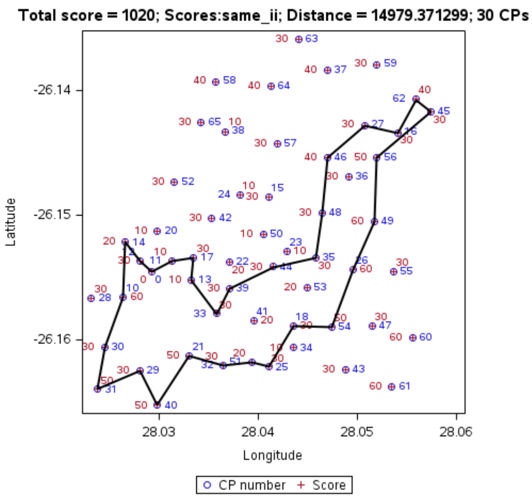
(h) 12 km



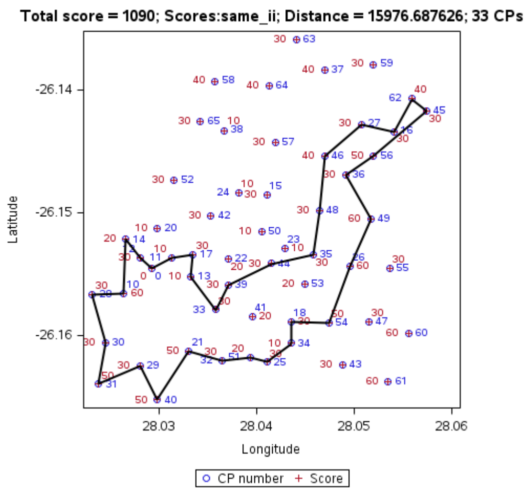
(i) 13 km



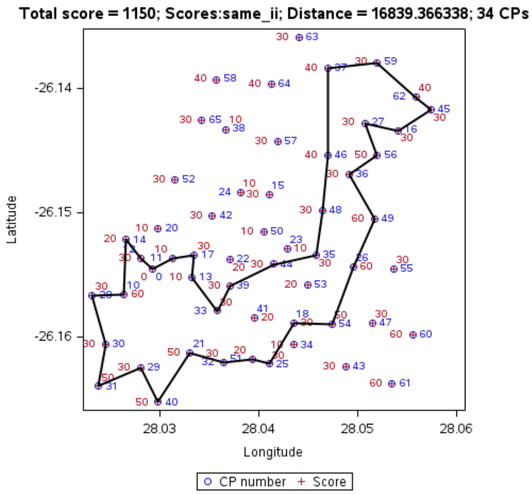
(j) 14 km



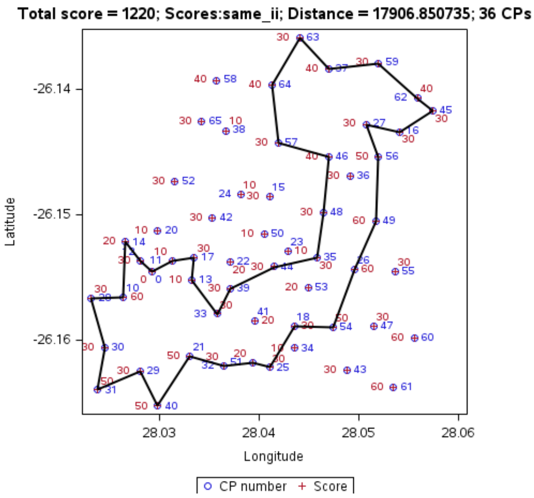
(k) 15 km



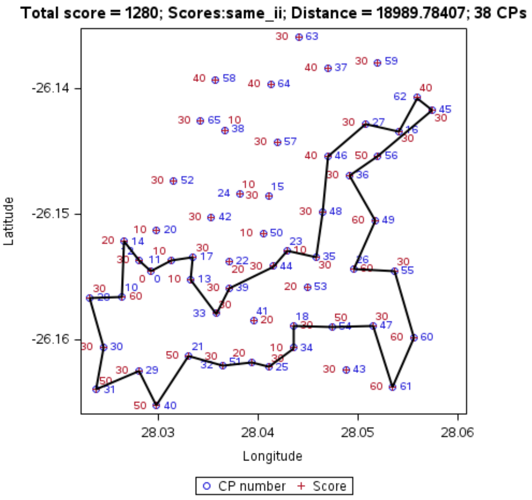
(l) 16 km



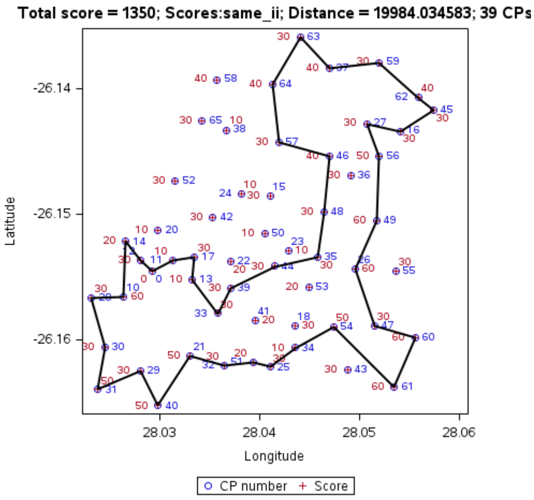
(m) 17 km



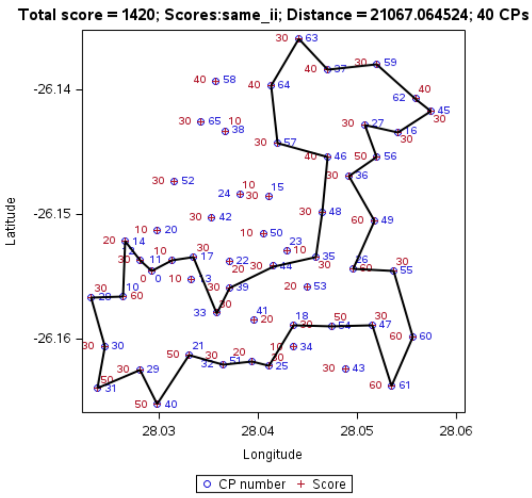
(n) 18 km



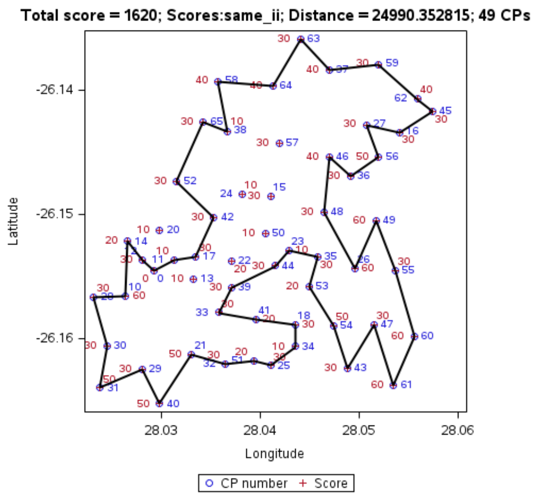
(o) 19 km



(p) 20 km

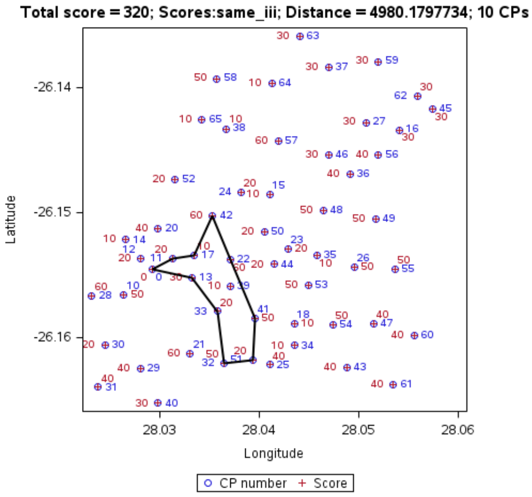


(q) 21,1 km

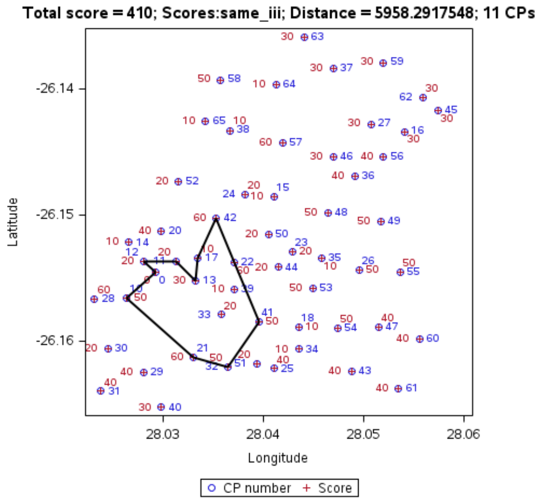


(r) 25 km

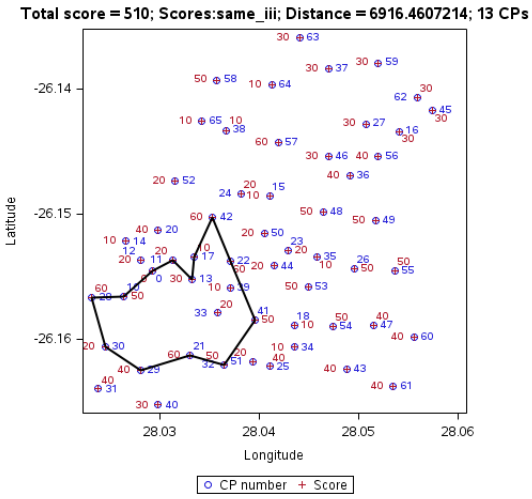
K.4 SAME_III OP output



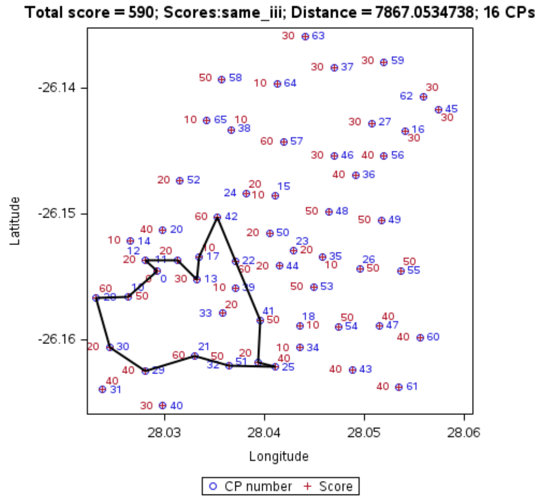
(a) 5 km



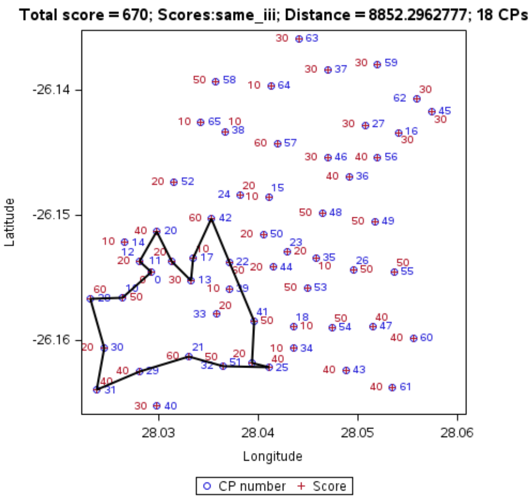
(b) 6 km



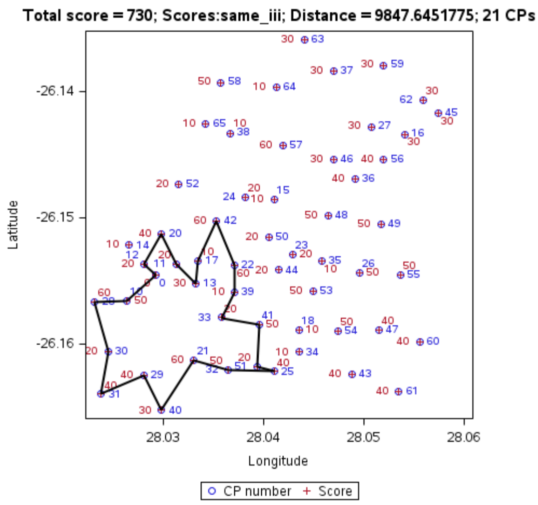
(c) 7 km



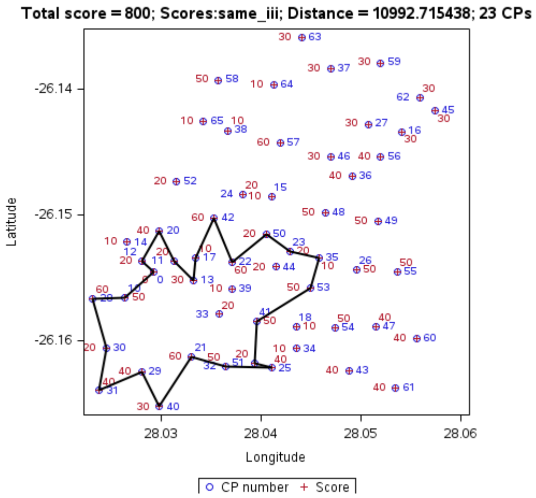
(d) 8 km



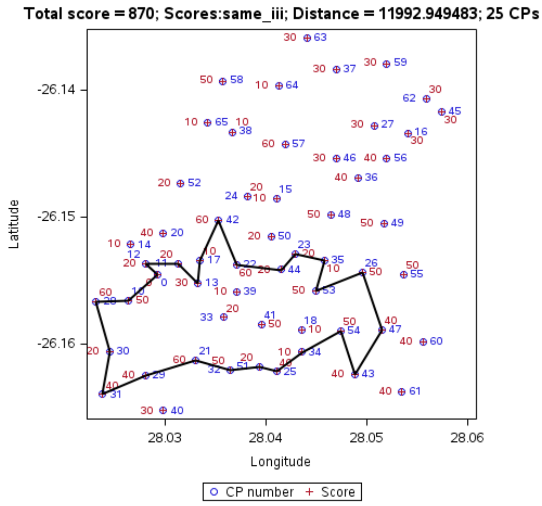
(e) 9 km



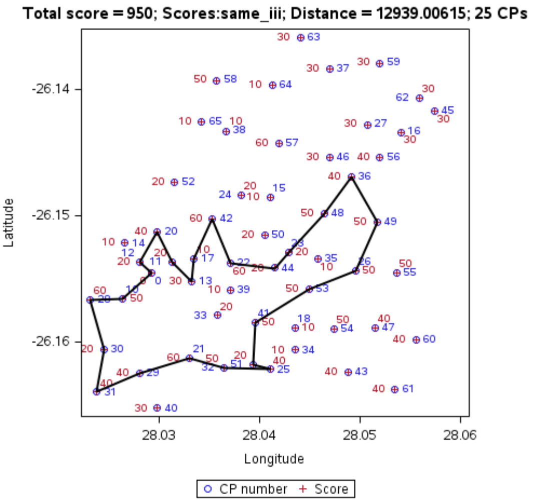
(f) 10 km



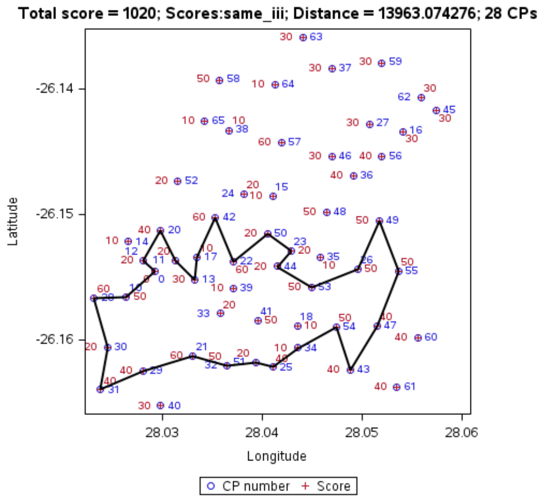
(g) 11 km



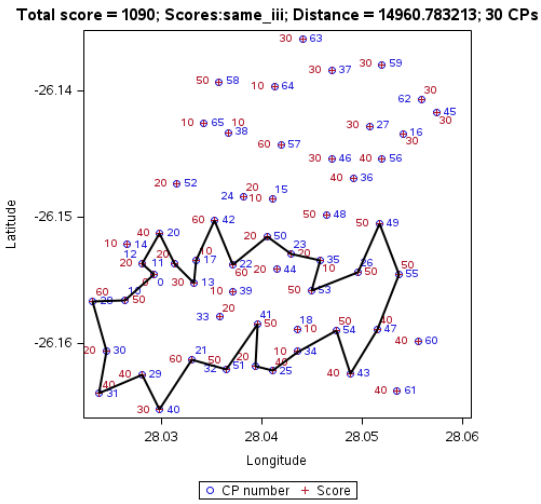
(h) 12 km



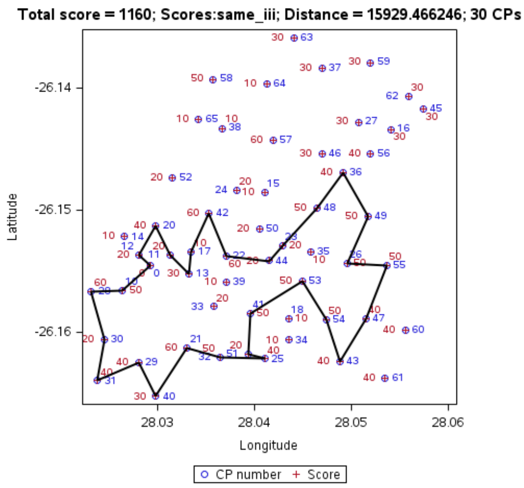
(i) 13 km



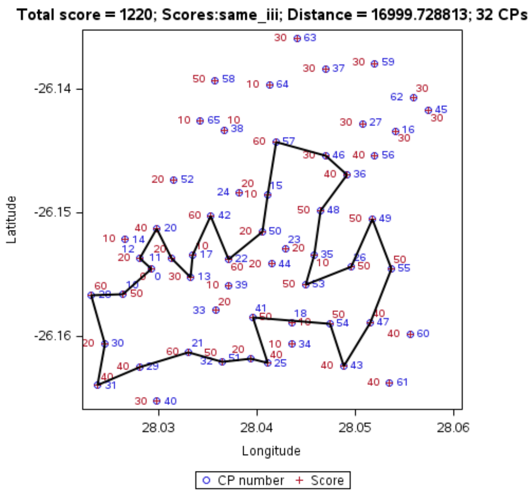
(j) 14 km



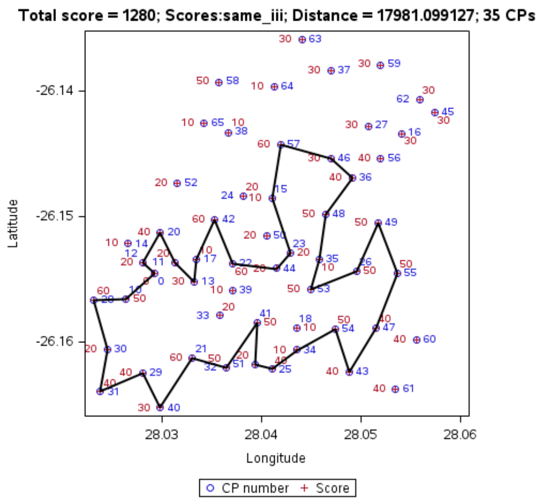
(k) 15 km



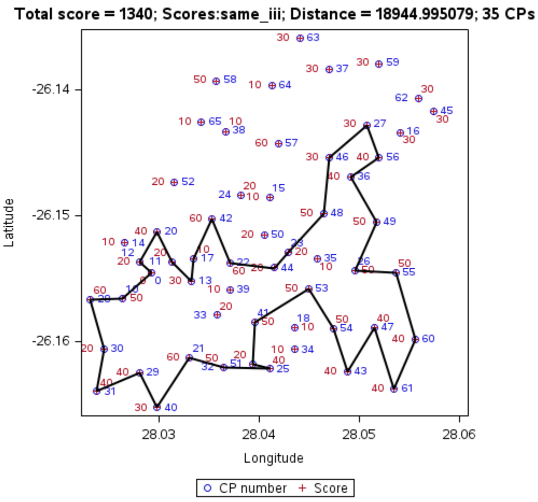
(l) 16 km



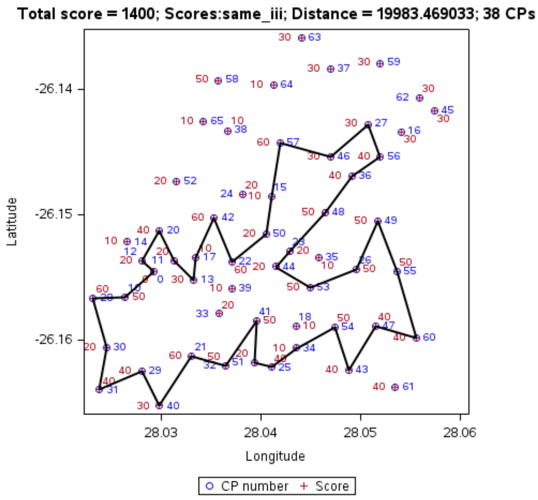
(m) 17 km



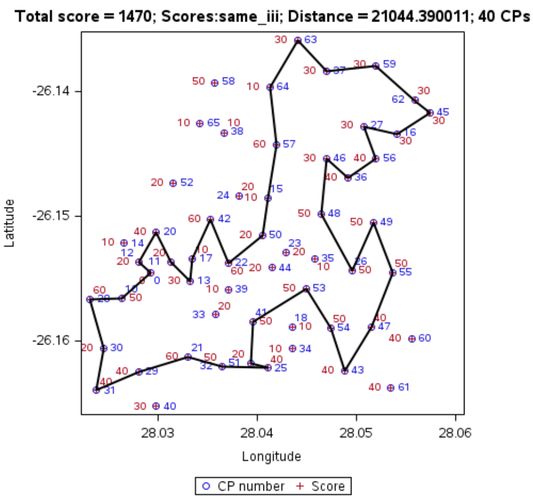
(n) 18 km



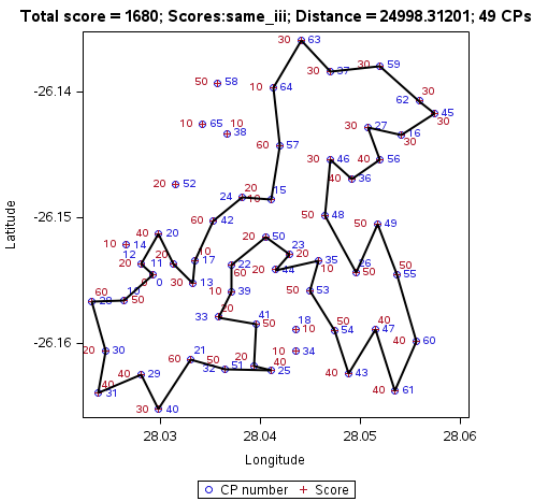
(o) 19 km



(p) 20 km

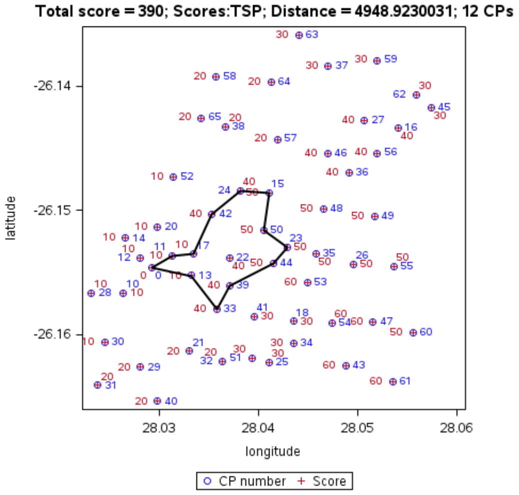


(q) 21,1 km

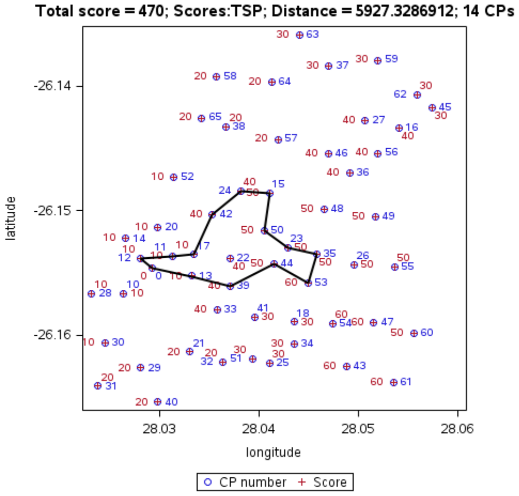


(r) 25 km

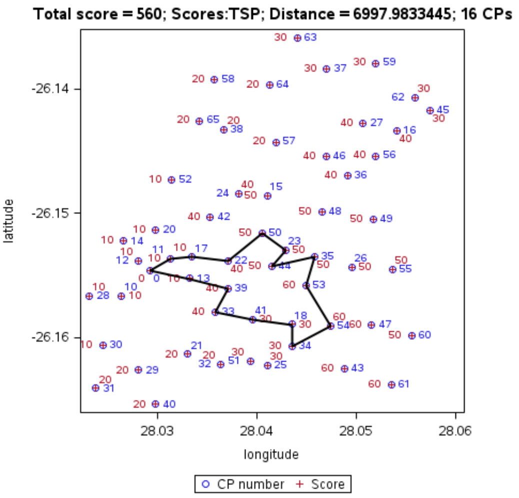
K.5 TSP OP output



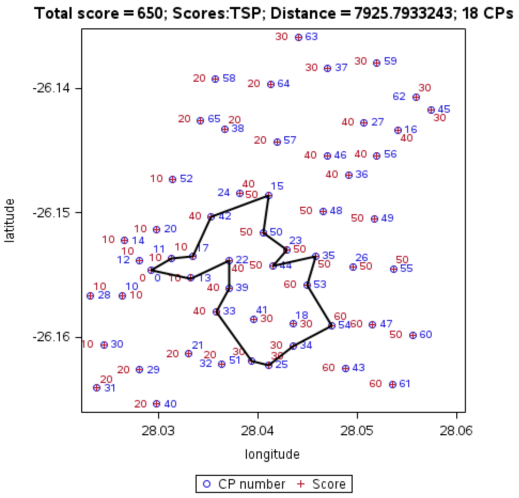
(a) 5 km



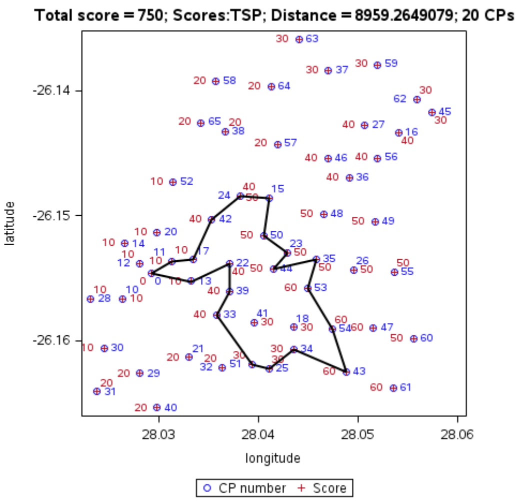
(b) 6 km



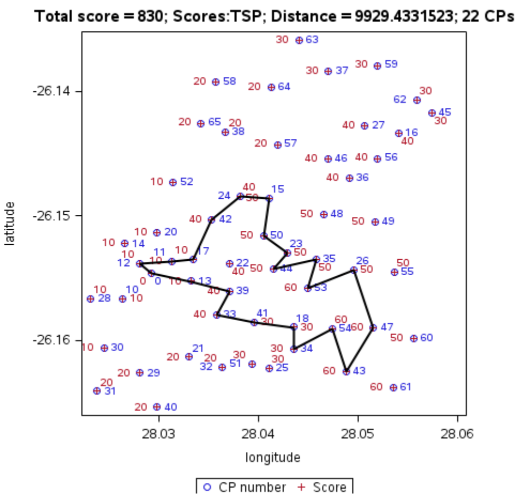
(c) 7 km



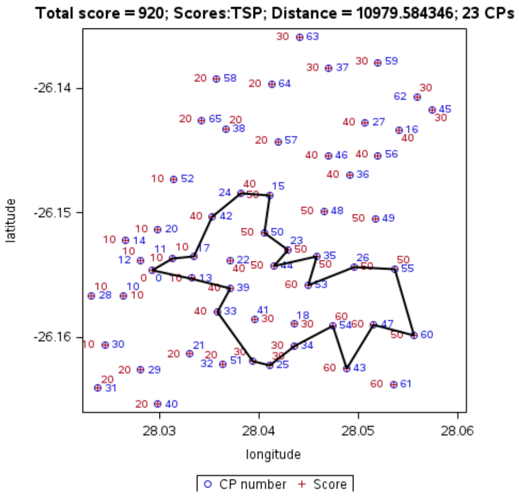
(d) 8 km



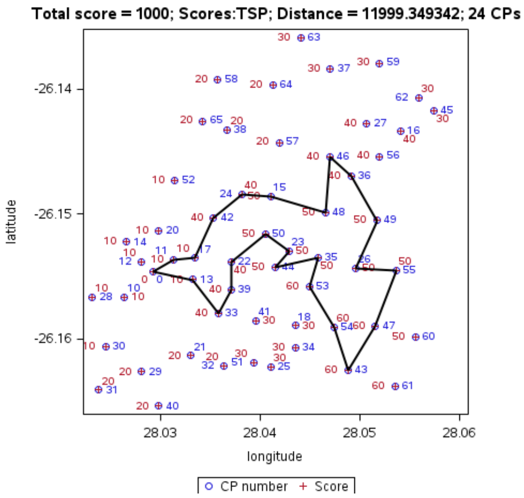
(e) 9 km



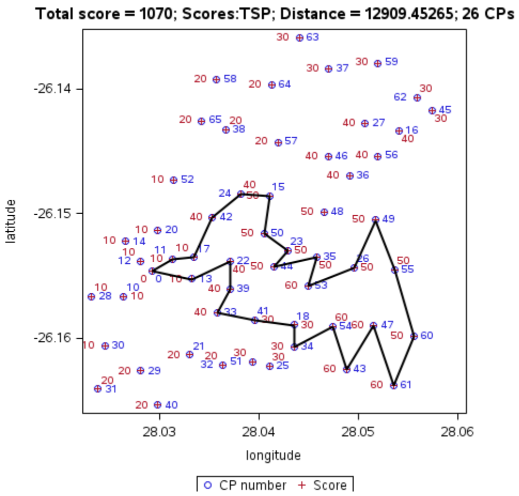
(f) 10 km



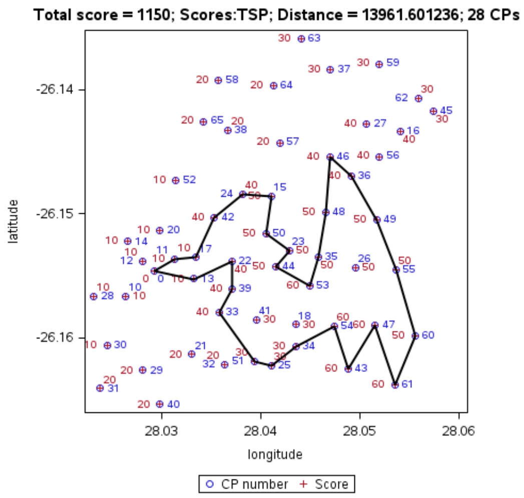
(g) 11 km



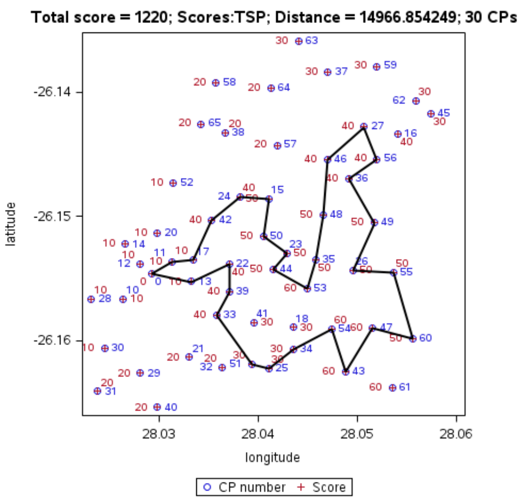
(h) 12 km



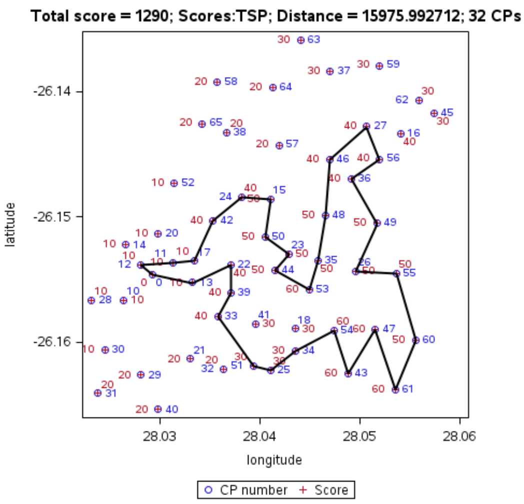
(i) 13 km



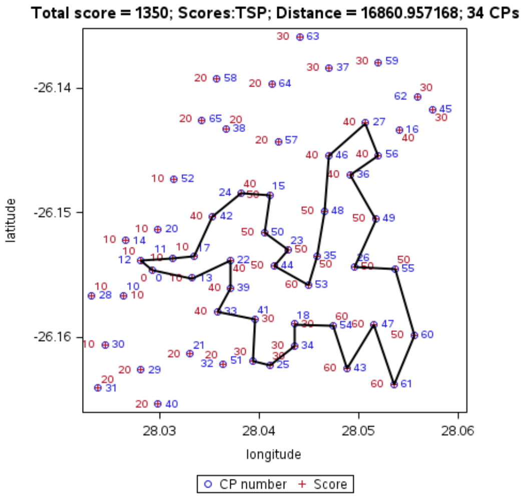
(j) 14 km



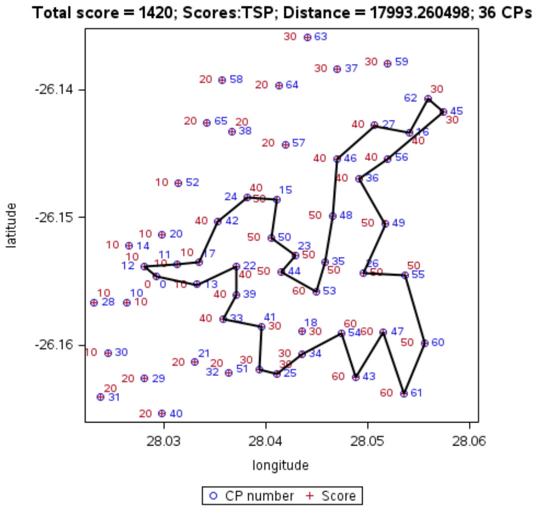
(k) 15 km



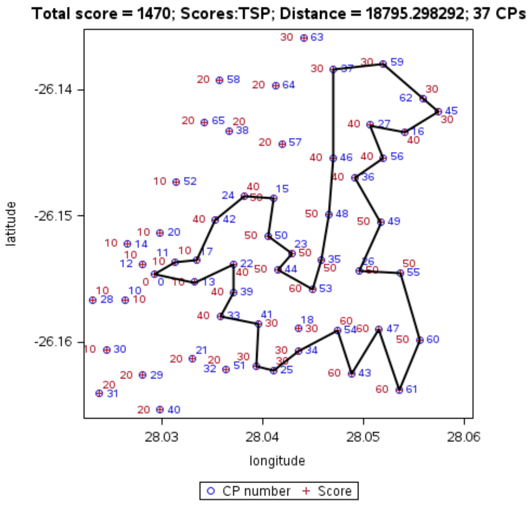
(l) 16 km



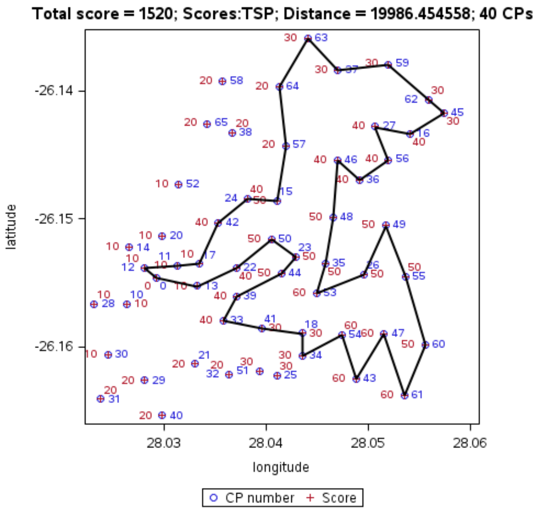
(m) 17 km



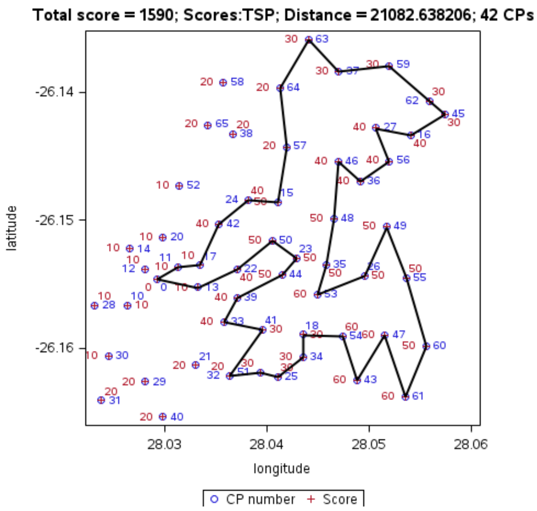
(n) 18 km



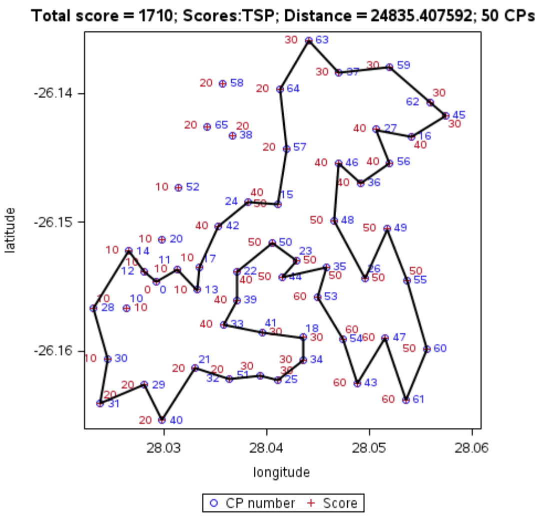
(o) 19 km



(p) 20 km

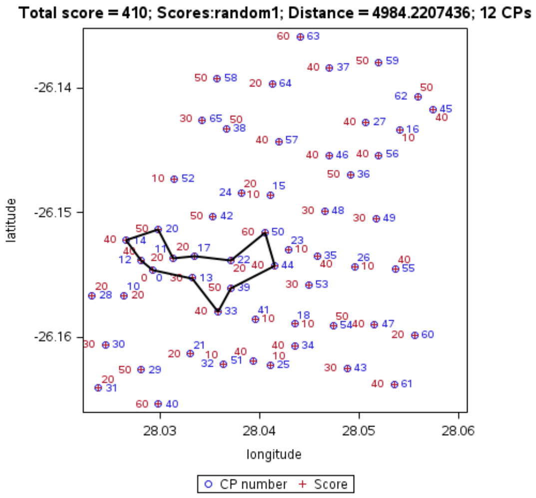


(q) 21,1 km

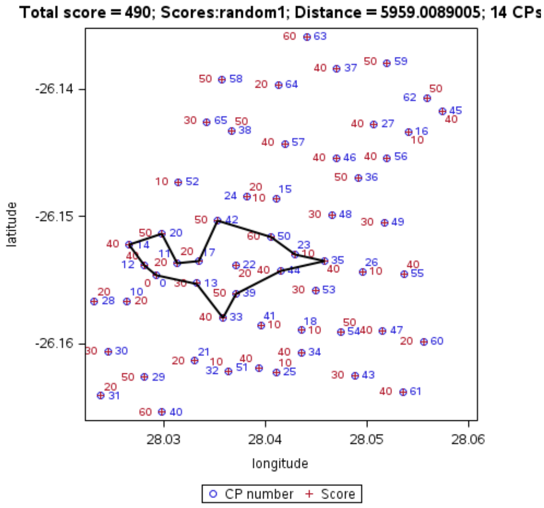


(r) 25 km

K.6 RANDOM OP output

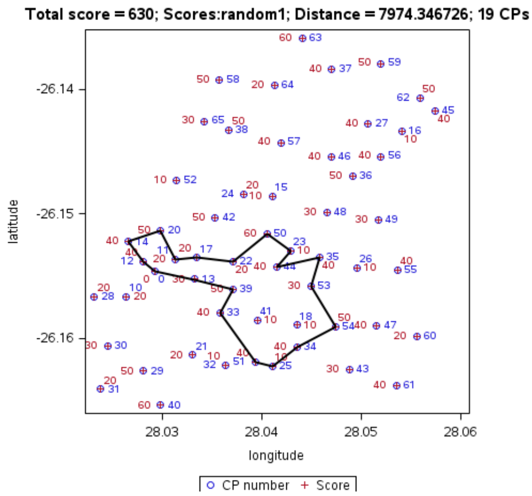


(a) 5 km

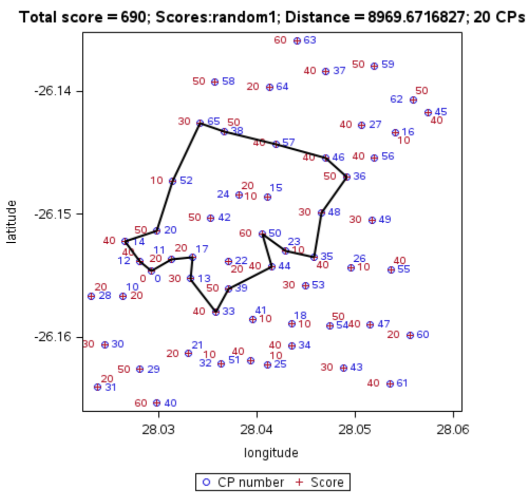


(b) 6 km

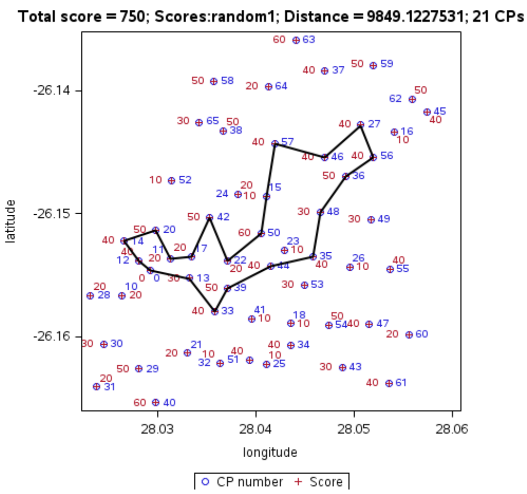
(c) 7 km



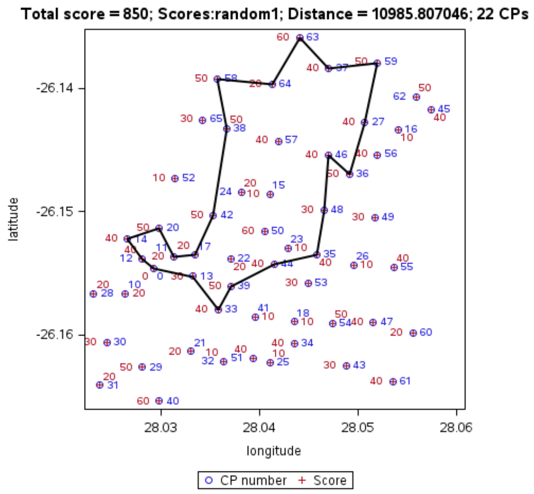
(d) 8 km



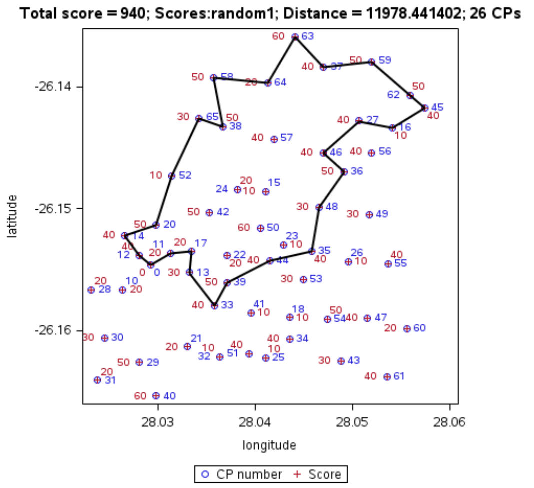
(e) 9 km



(f) 10 km

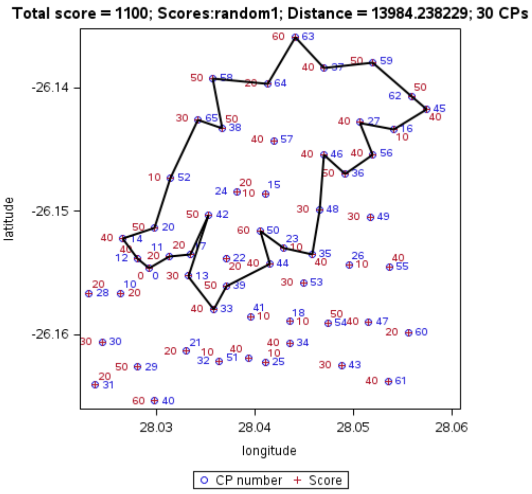


(g) 11 km

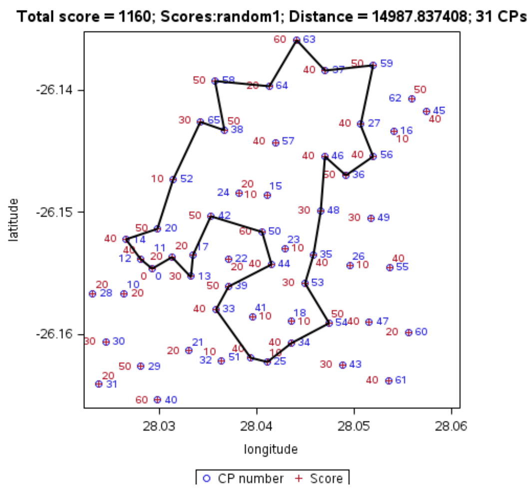


(h) 12 km

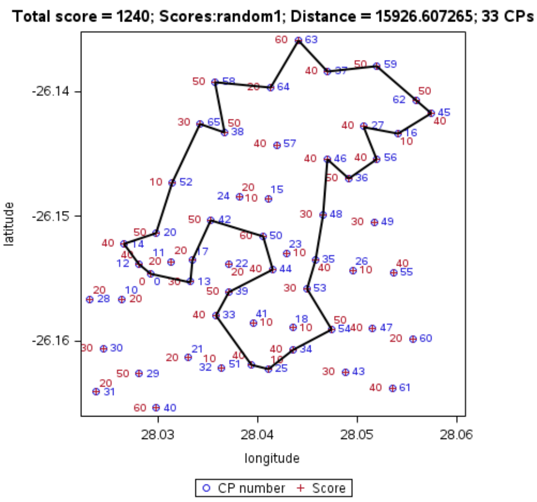
(i) 13 km



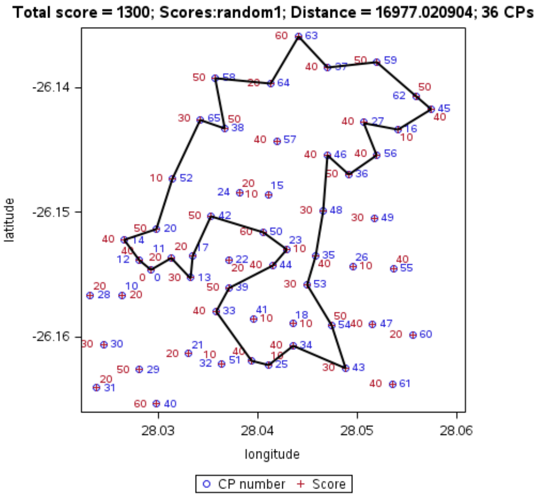
(j) 14 km



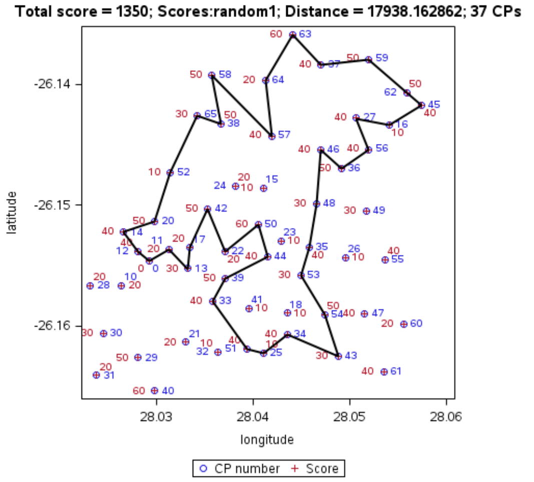
(k) 15 km



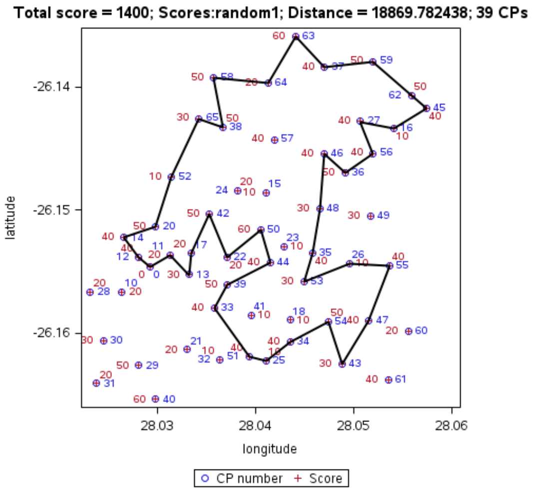
(l) 16 km



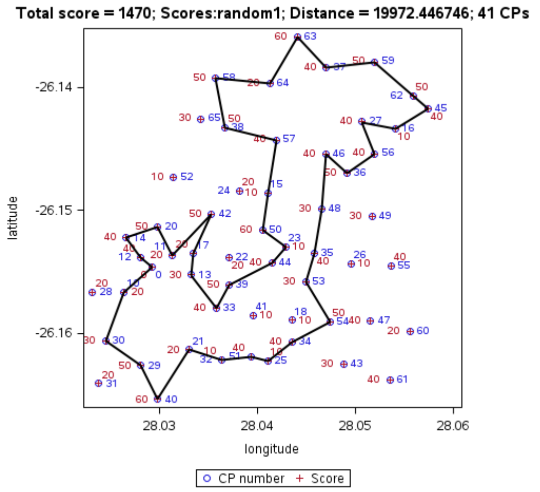
(m) 17 km



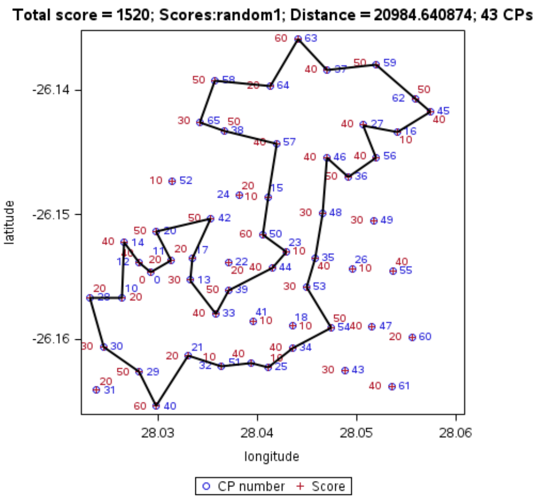
(n) 18 km



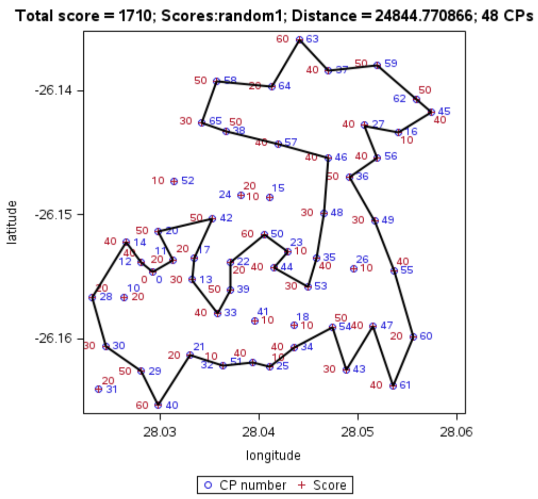
(o) 19 km



(p) 20 km

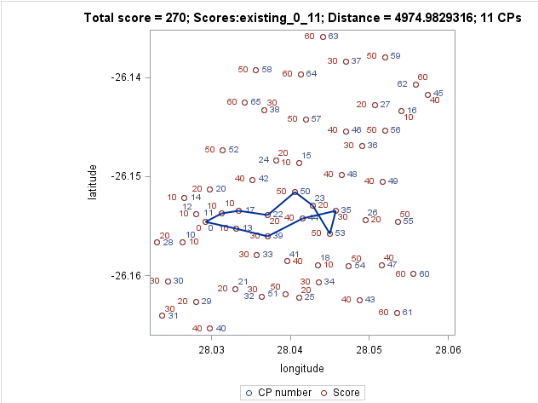


(q) 21,1 km

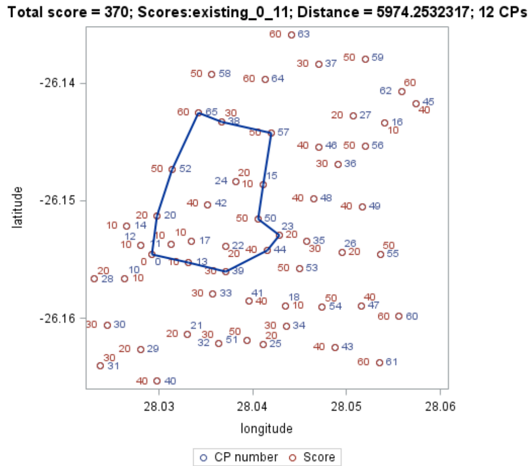


(r) 25 km

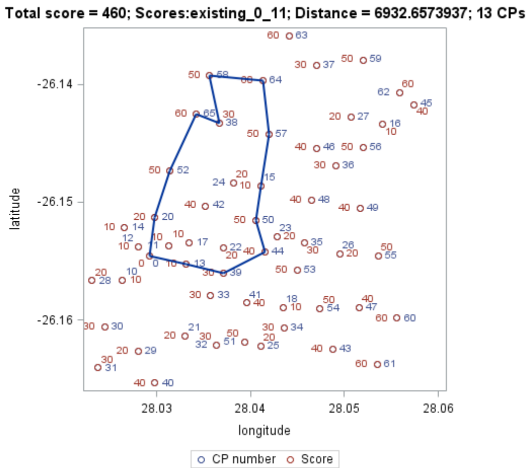
K.7 EXIST OP output



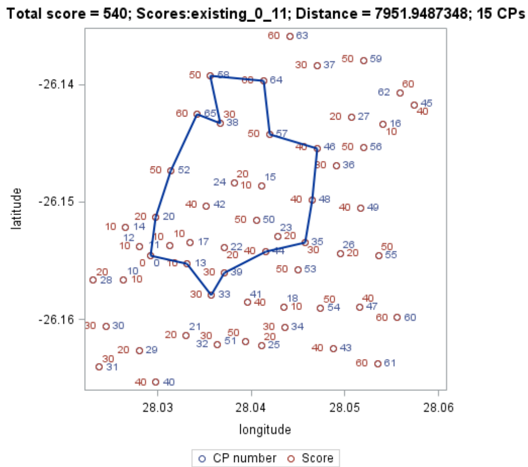
(a) 5 km



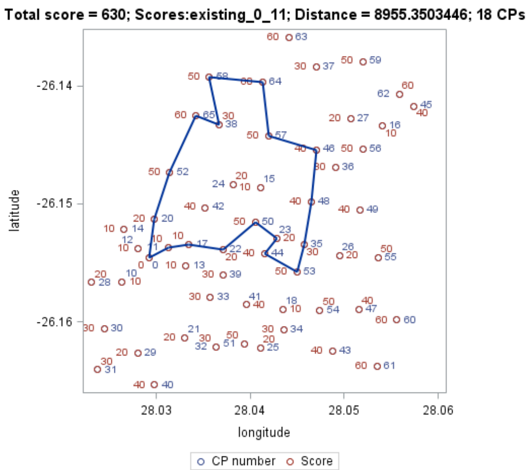
(b) 6 km



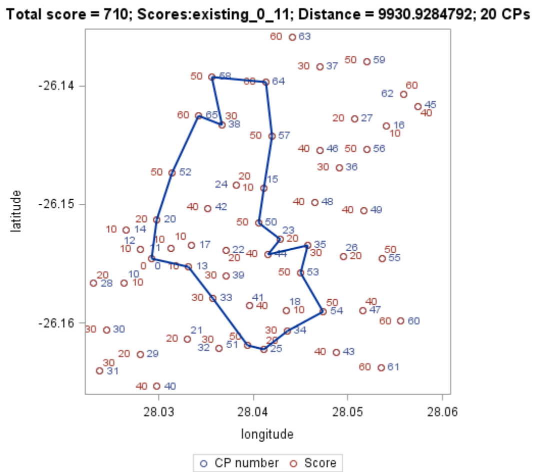
(c) 7 km



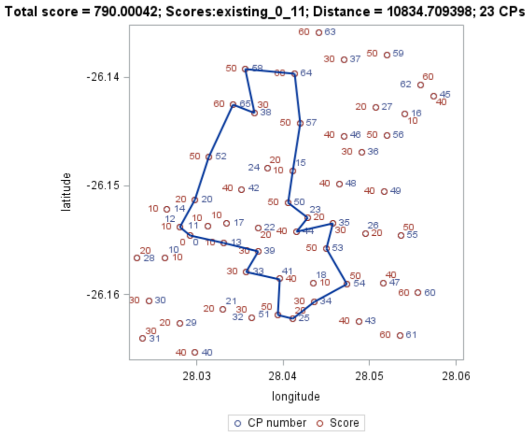
(d) 8 km



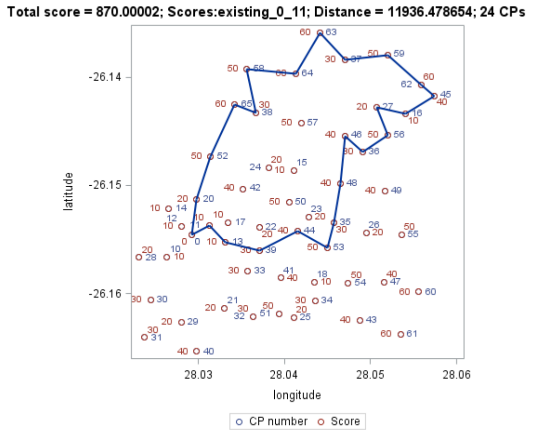
(e) 9 km



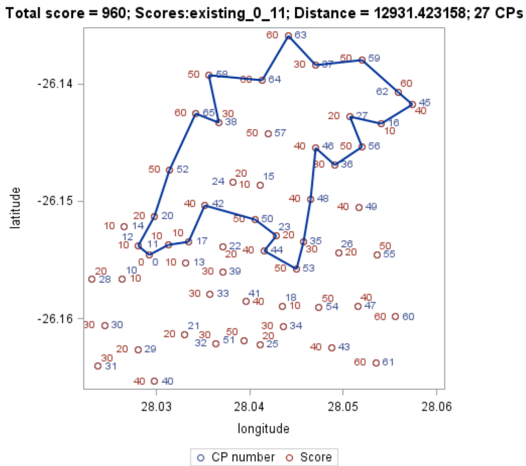
(f) 10 km



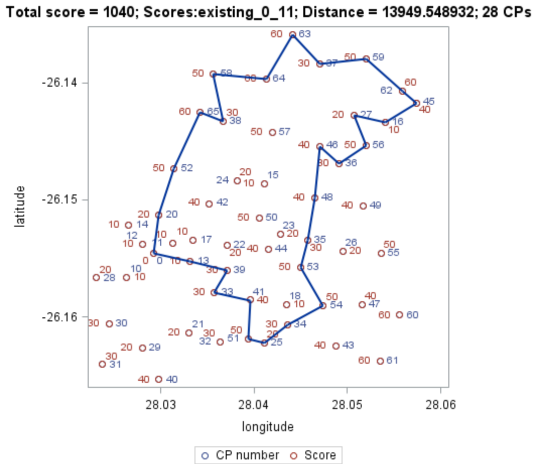
(g) 11 km



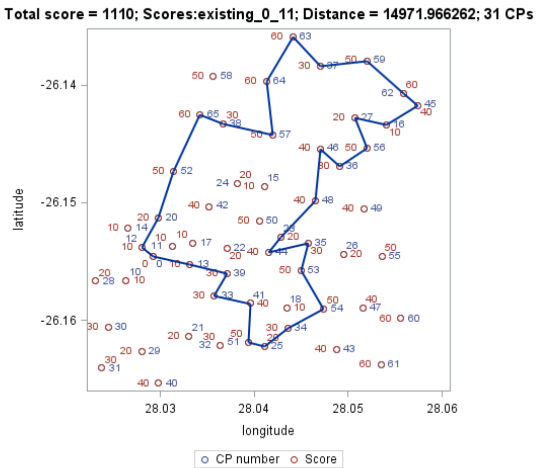
(h) 12 km



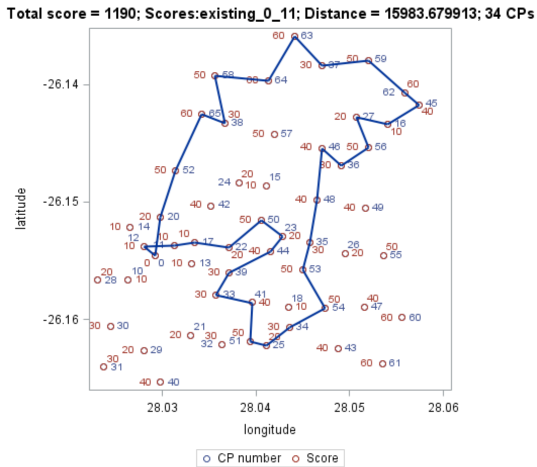
(i) 13 km



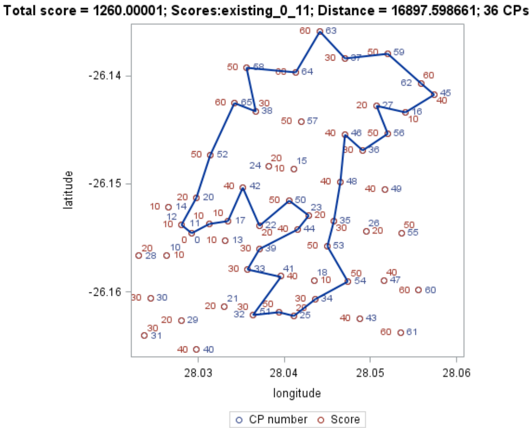
(j) 14 km



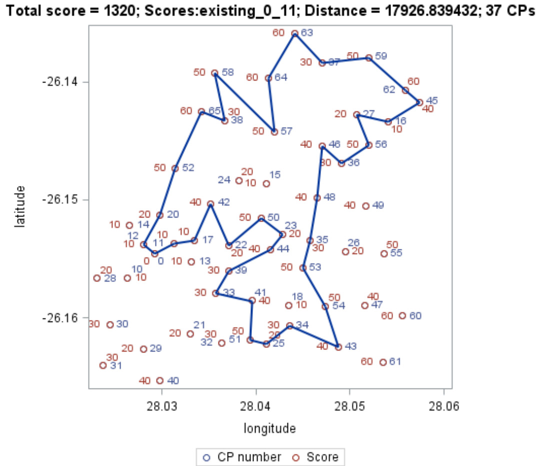
(k) 15 km



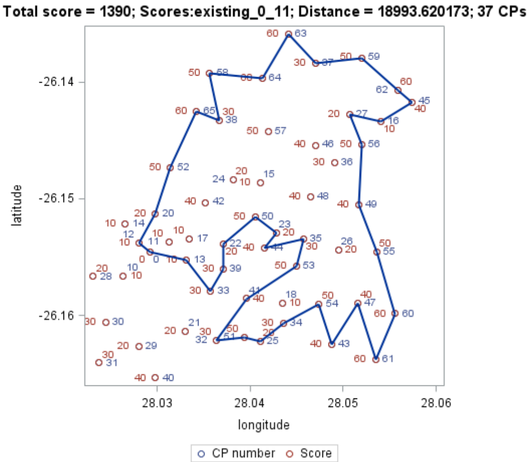
(l) 16 km



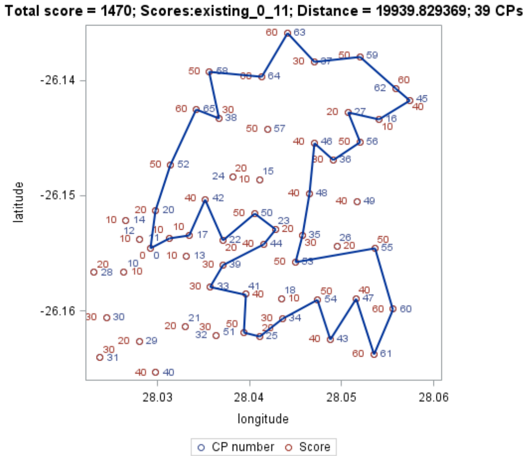
(m) 17 km



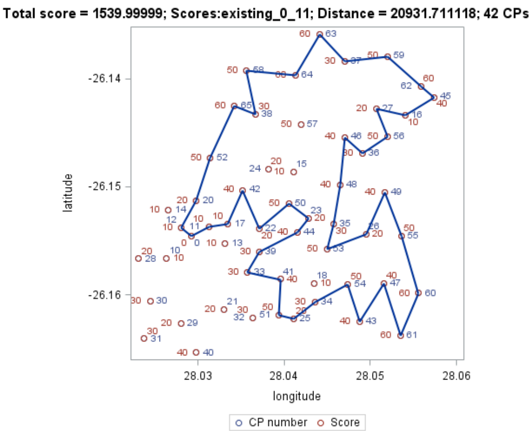
(n) 18 km



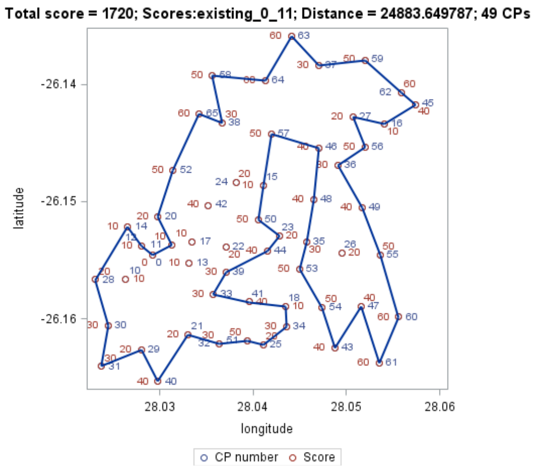
(o) 19 km



(p) 20 km

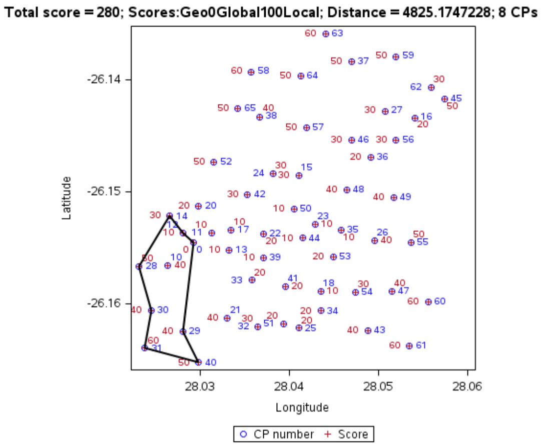


(q) 21,1 km

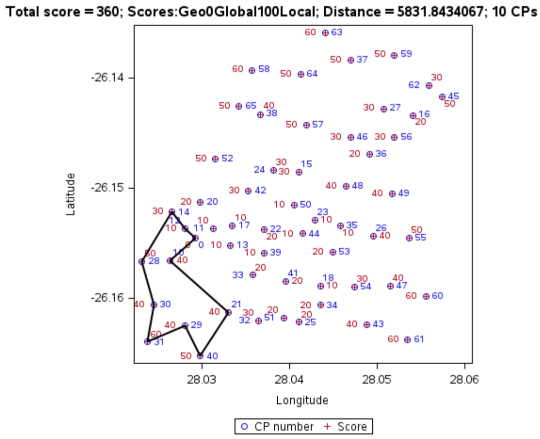


(r) 25 km

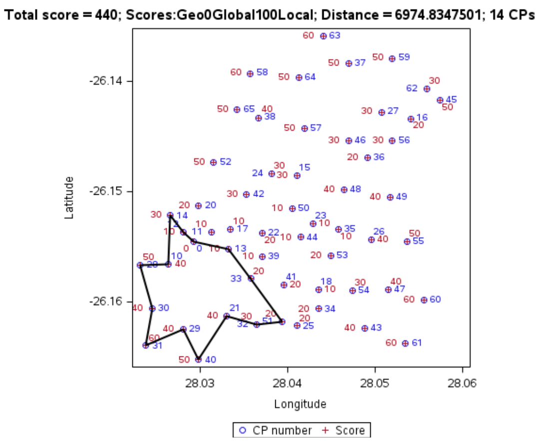
K.8 Geo0G100L OP output



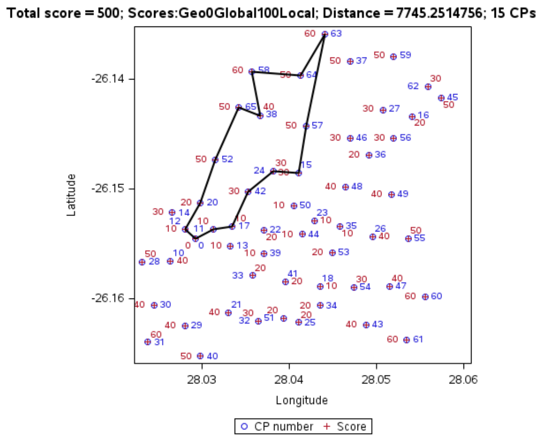
(a) 5 km



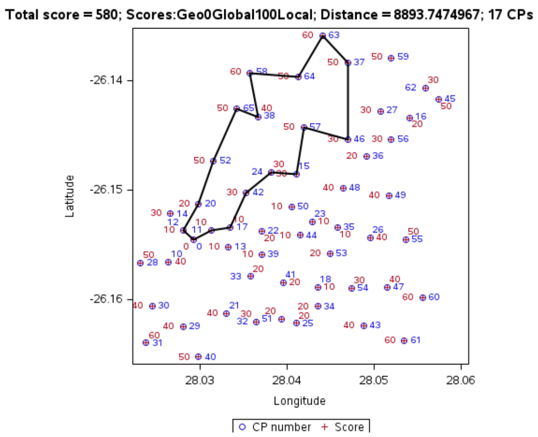
(b) 6 km



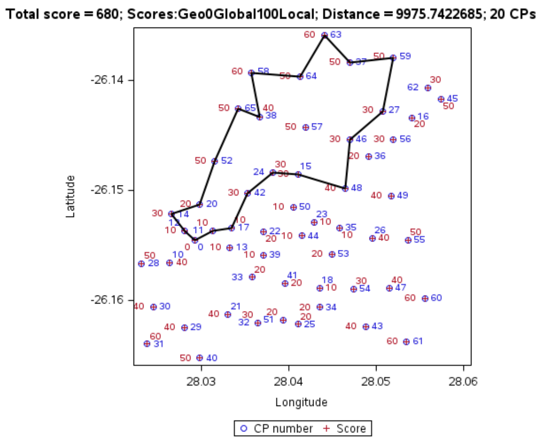
(c) 7 km



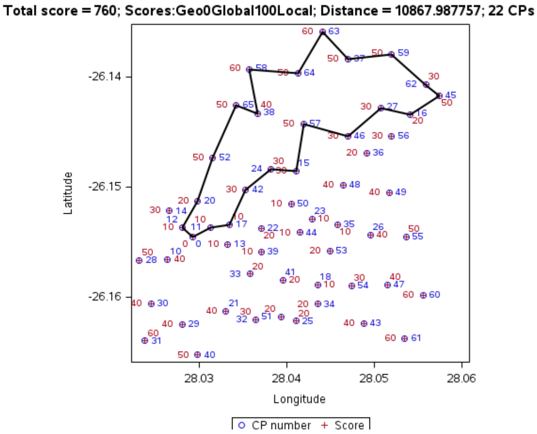
(d) 8 km



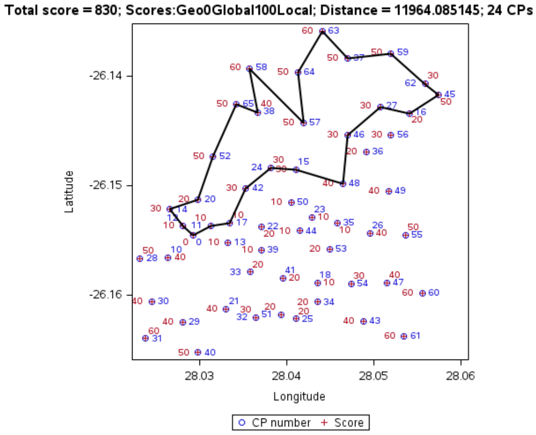
(e) 9 km



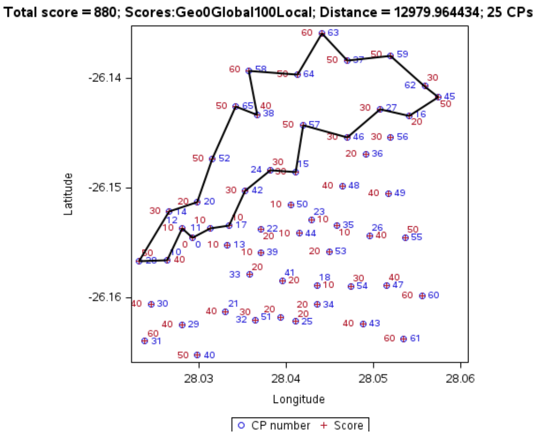
(f) 10 km



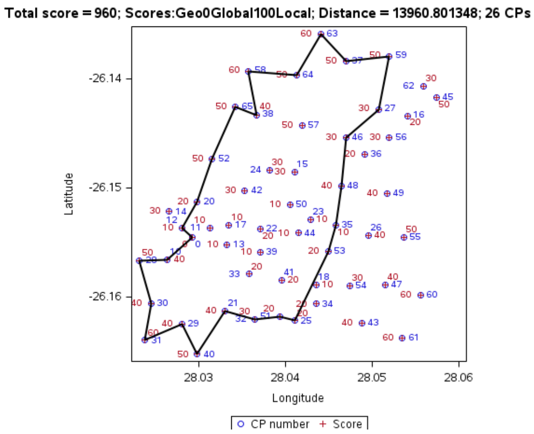
(g) 11 km



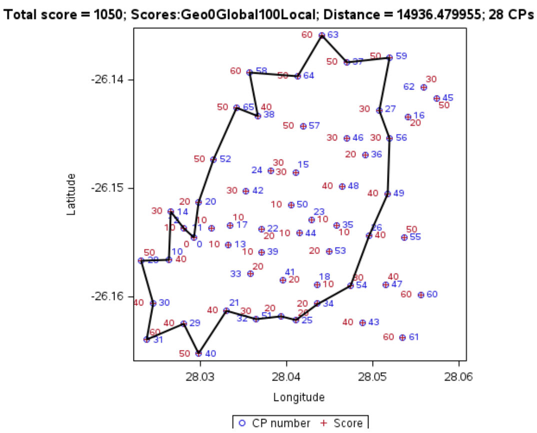
(h) 12 km



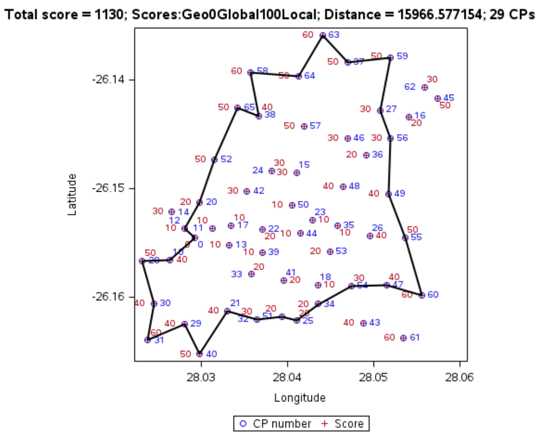
(i) 13 km



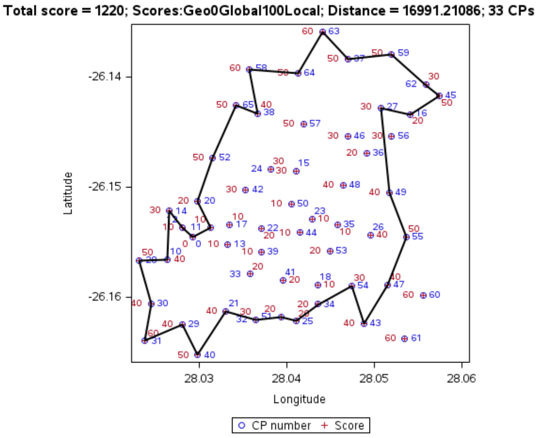
(j) 14 km



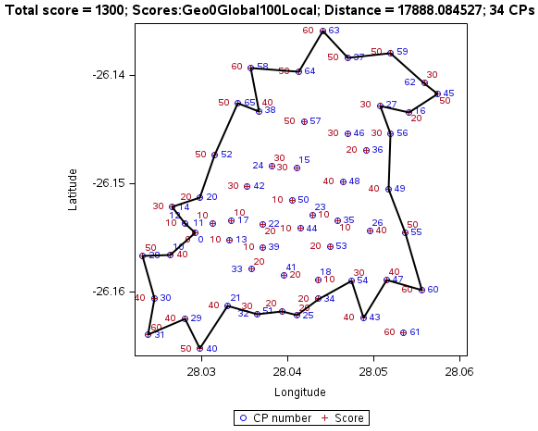
(k) 15 km



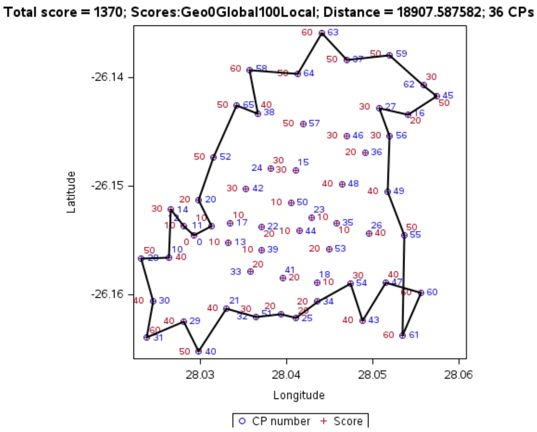
(l) 16 km



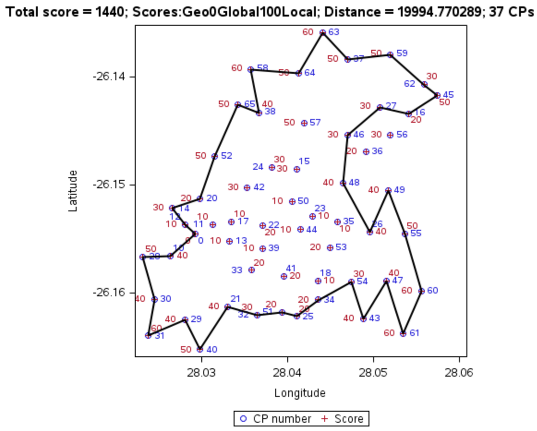
(m) 17 km



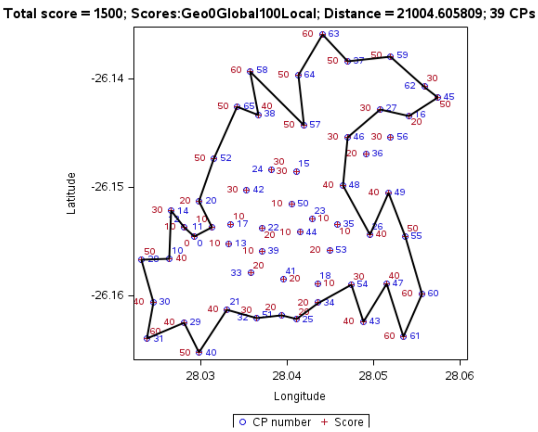
(n) 18 km



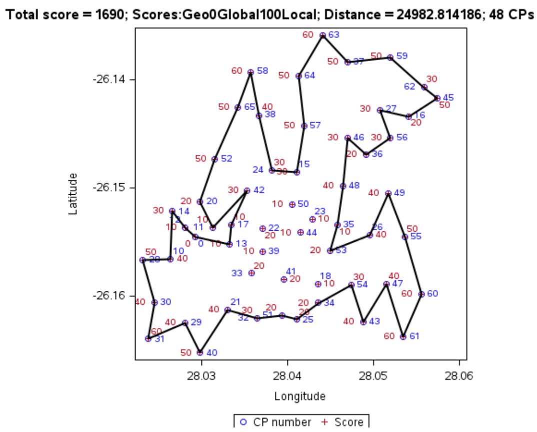
(o) 19 km



(p) 20 km

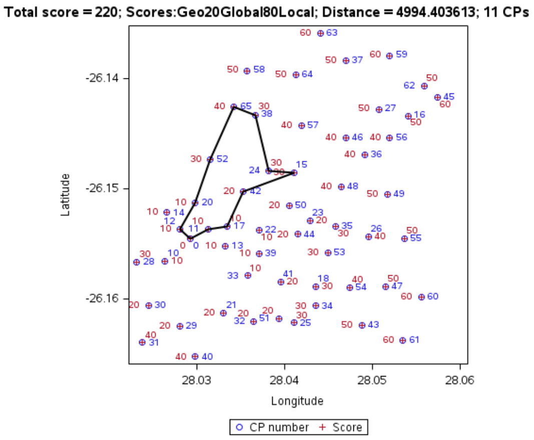


(q) 21,1 km

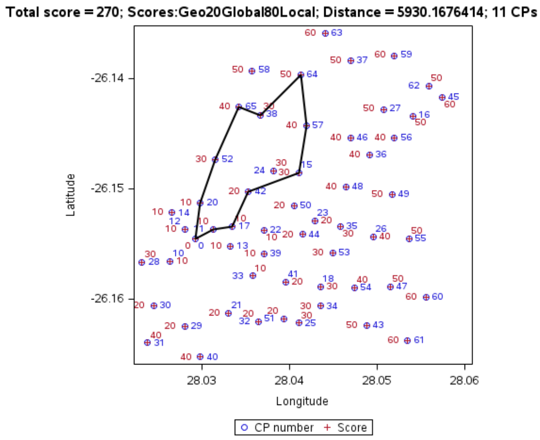


(r) 25 km

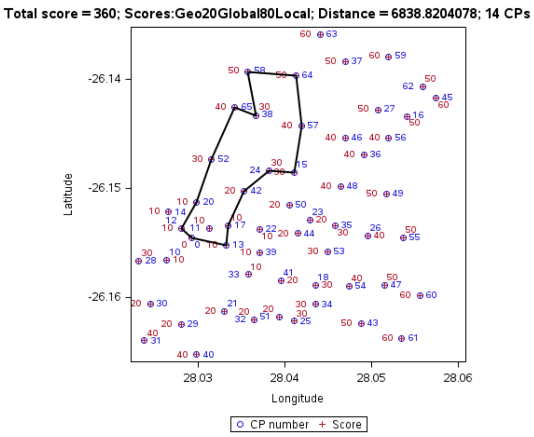
K.9 Geo20G80L OP output



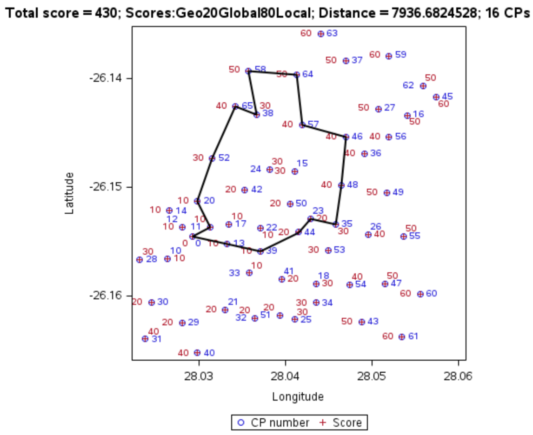
(a) 5 km



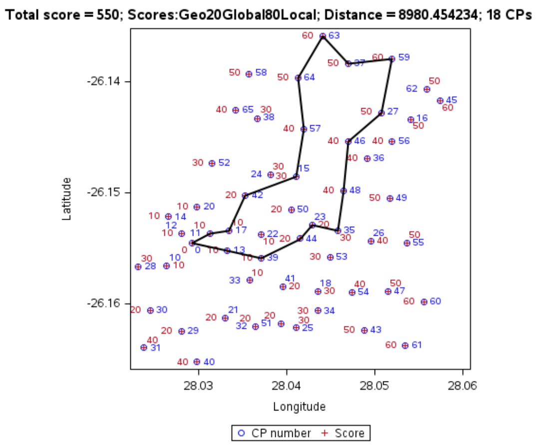
(b) 6 km



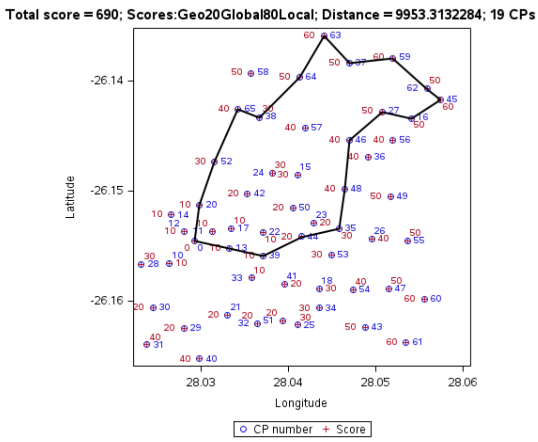
(c) 7 km



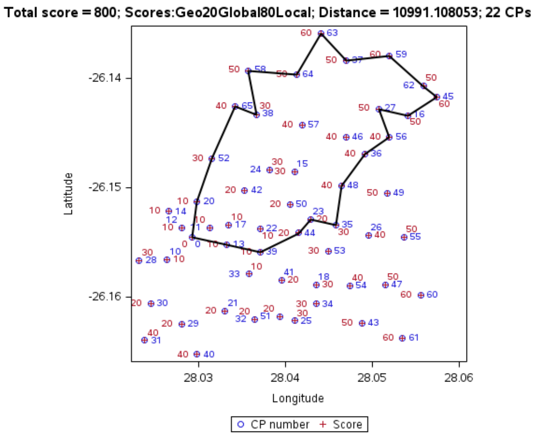
(d) 8 km



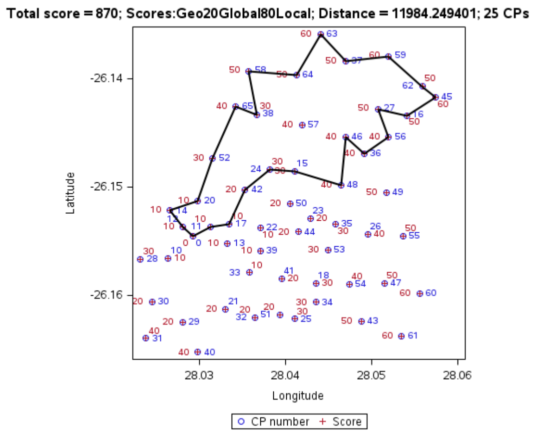
(e) 9 km



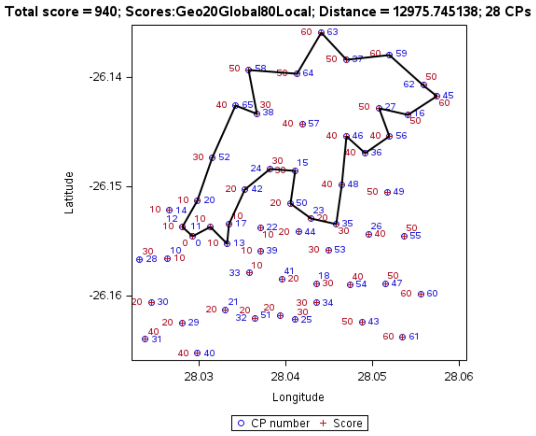
(f) 10 km



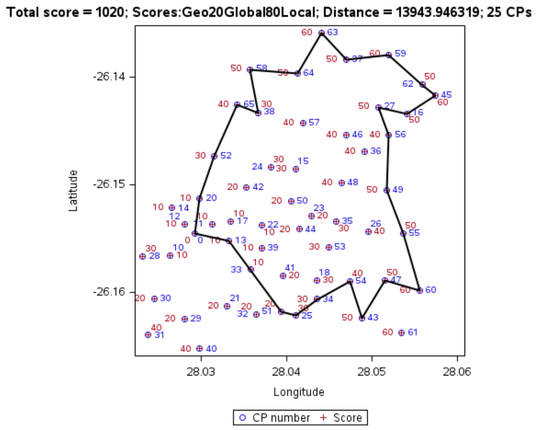
(g) 11 km



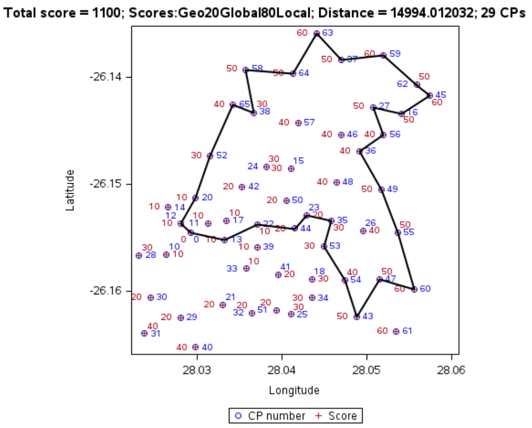
(h) 12 km



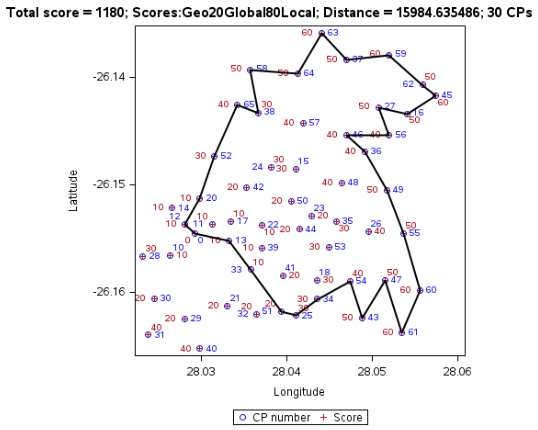
(i) 13 km



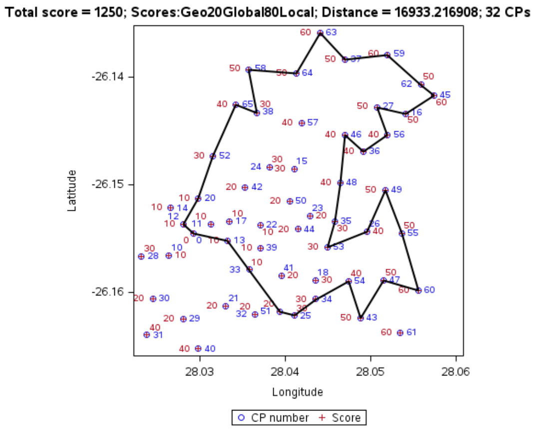
(j) 14 km



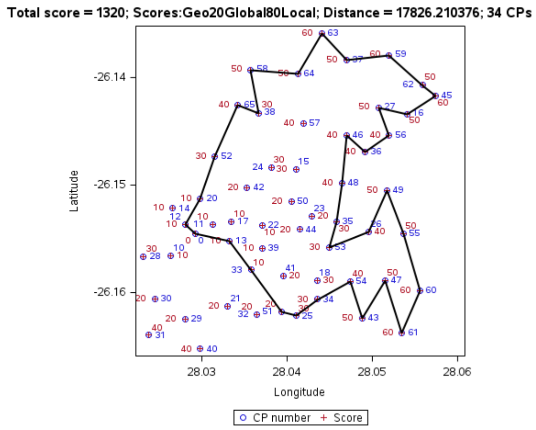
(k) 15 km



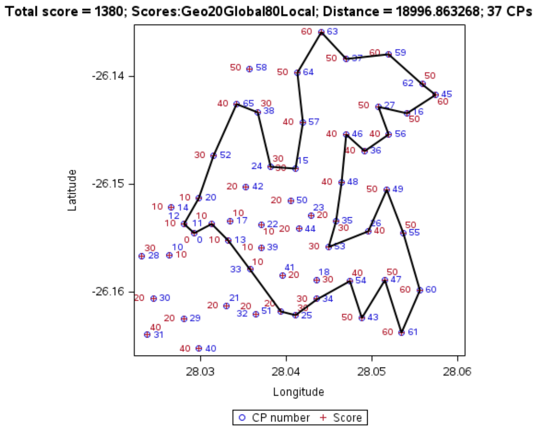
(l) 16 km



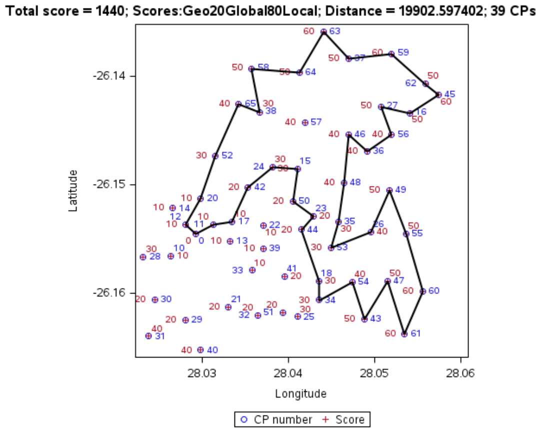
(m) 17 km



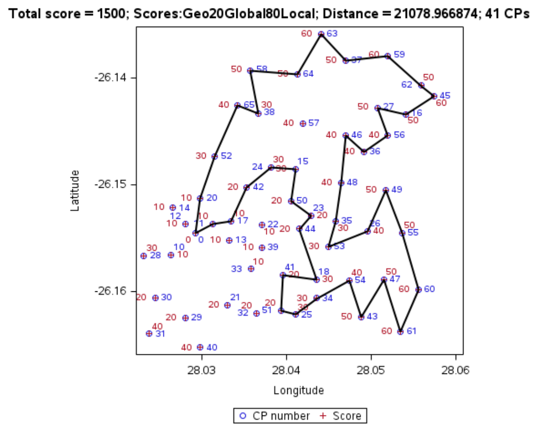
(n) 18 km



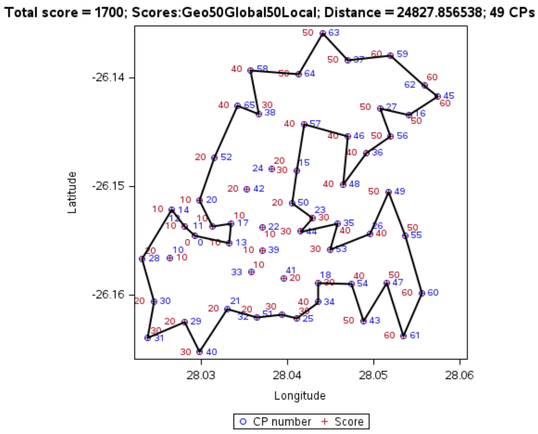
(o) 19 km



(p) 20 km

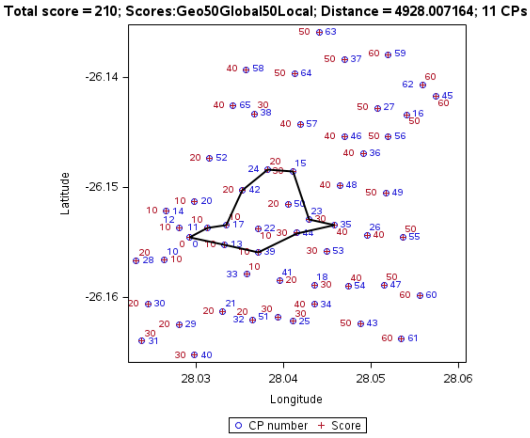


(q) 21,1 km

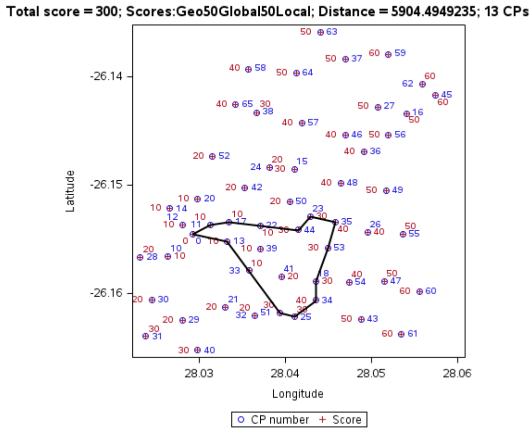


(r) 25 km

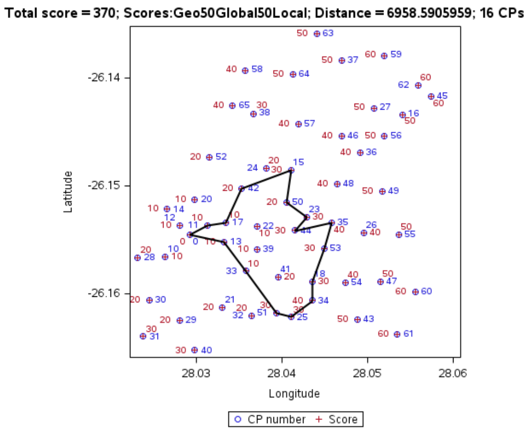
K.10 Geo50G50L OP output



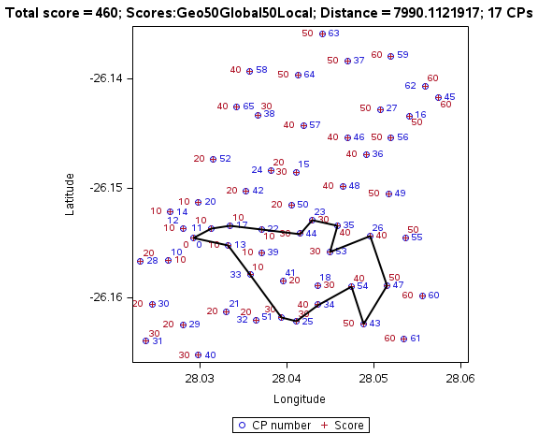
(a) 5 km



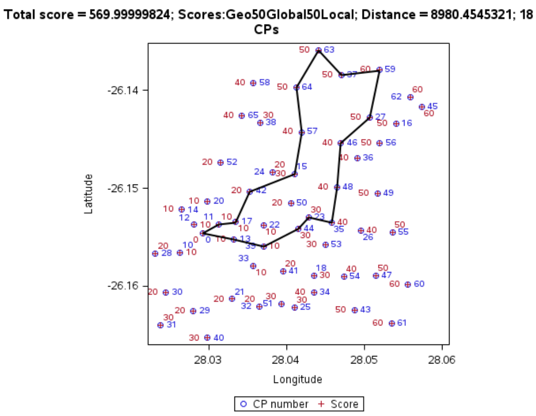
(b) 6 km



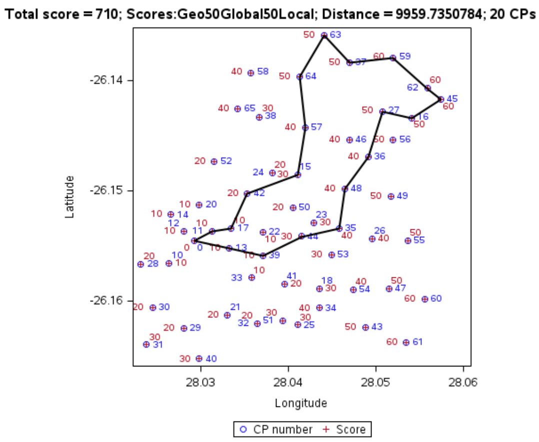
(c) 7 km



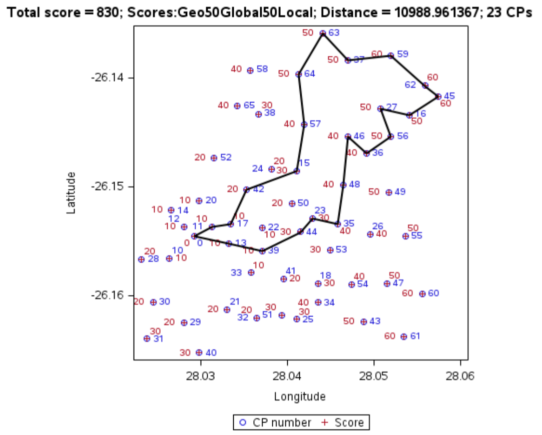
(d) 8 km



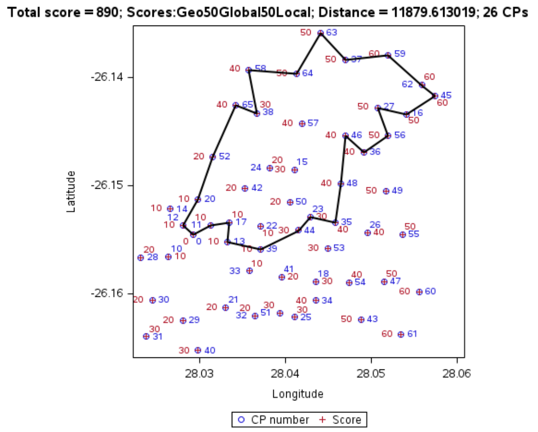
(e) 9 km



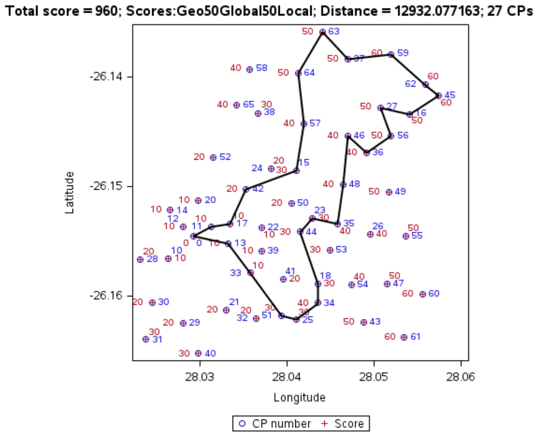
(f) 10 km



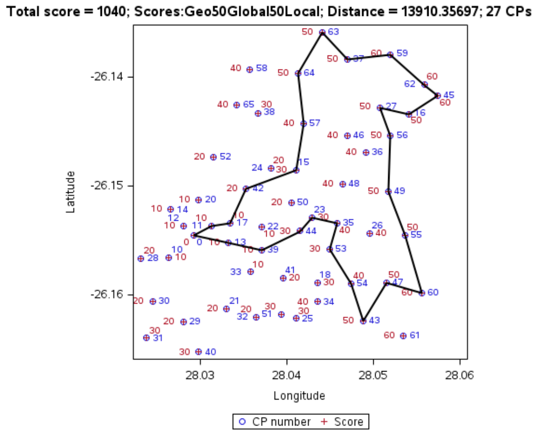
(g) 11 km



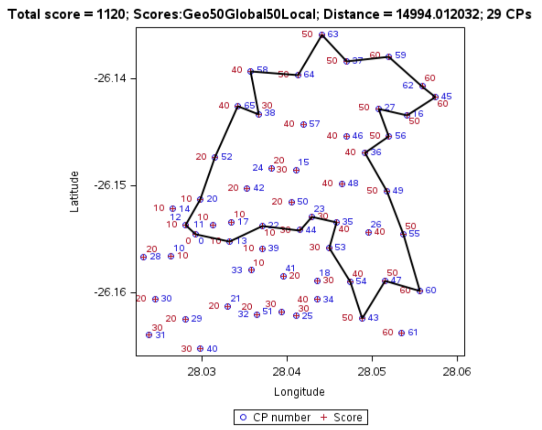
(h) 12 km



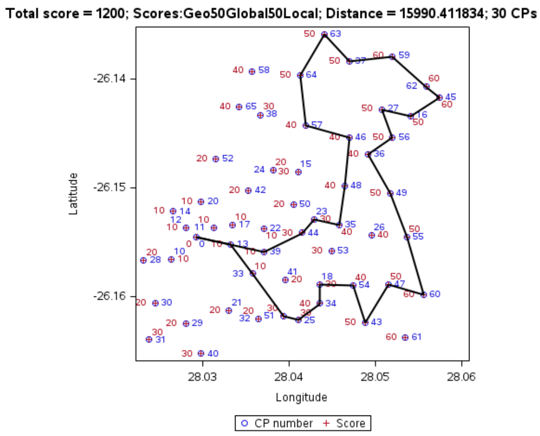
(i) 13 km



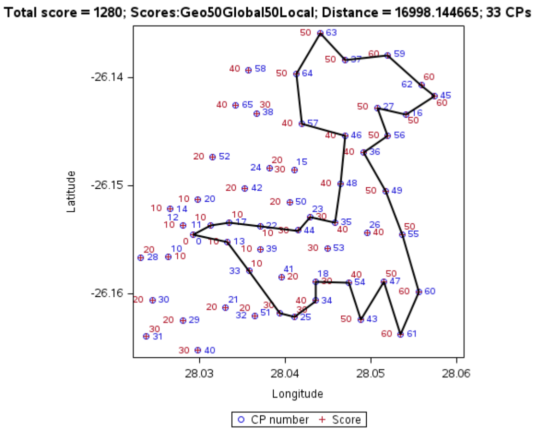
(j) 14 km



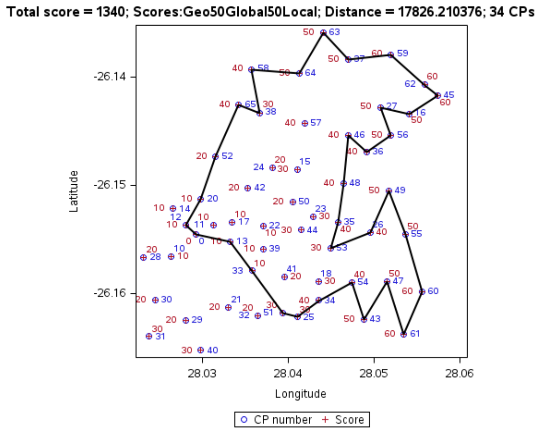
(k) 15 km



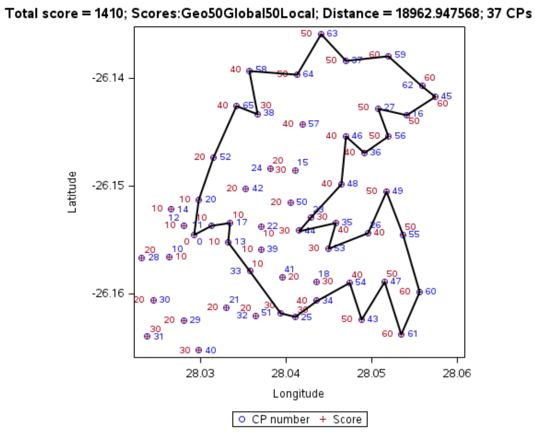
(l) 16 km



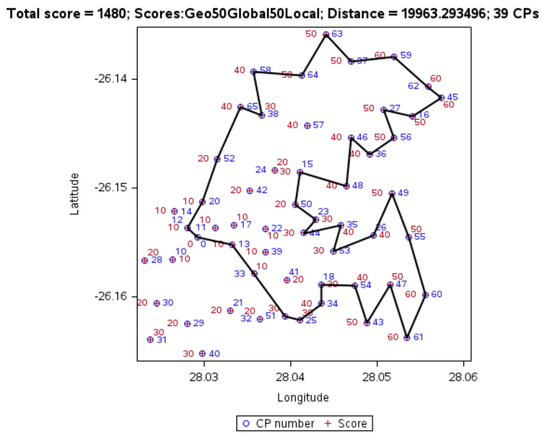
(m) 17 km



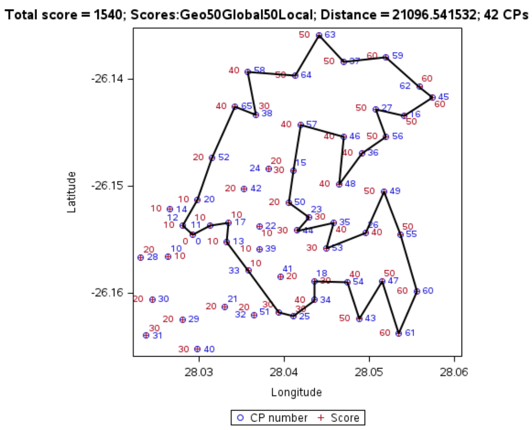
(n) 18 km



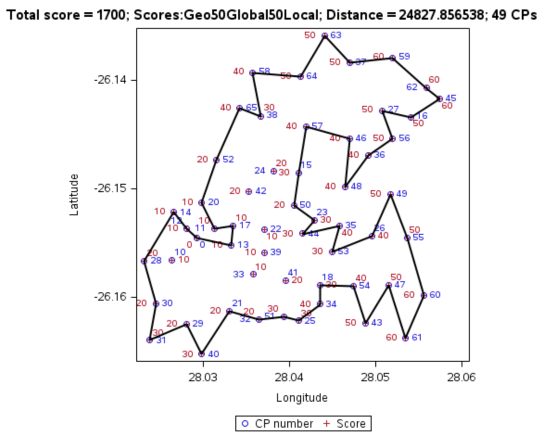
(o) 19 km



(p) 20 km

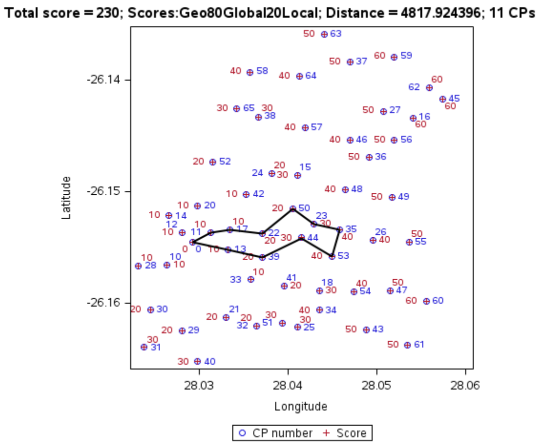


(q) 21,1 km

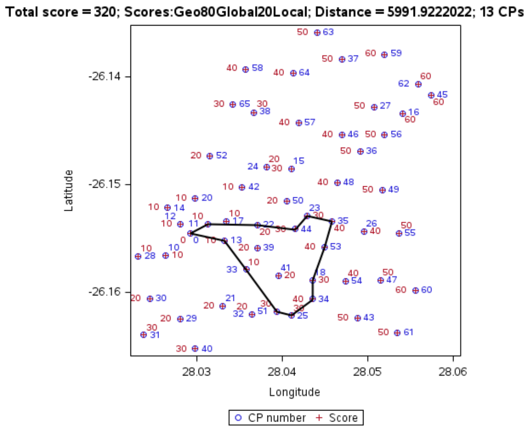


(r) 25 km

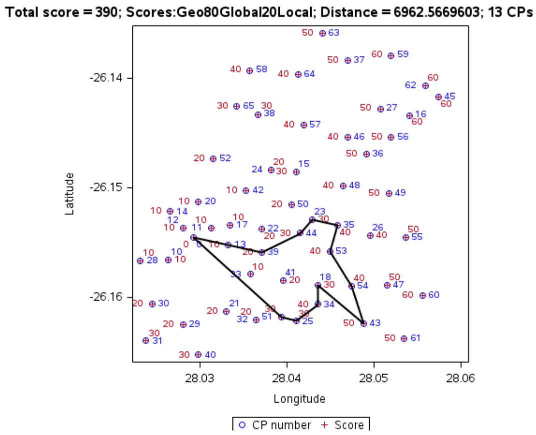
K.11 Geo80G20L OP output



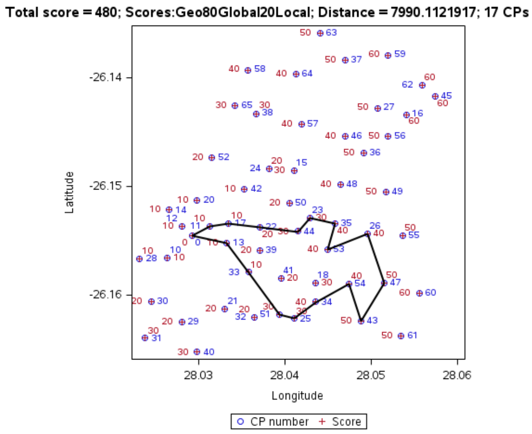
(a) 5 km



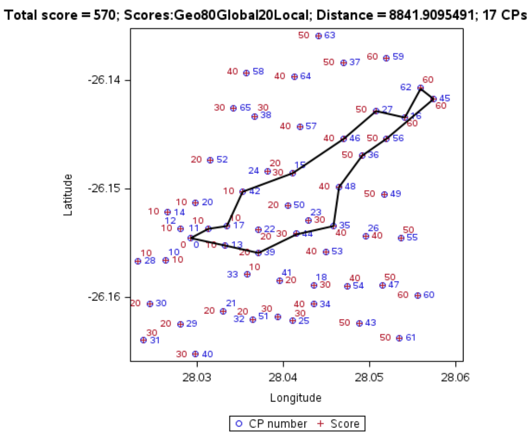
(b) 6 km



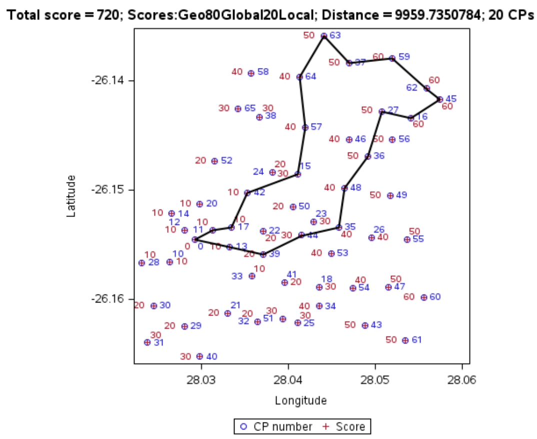
(c) 7 km



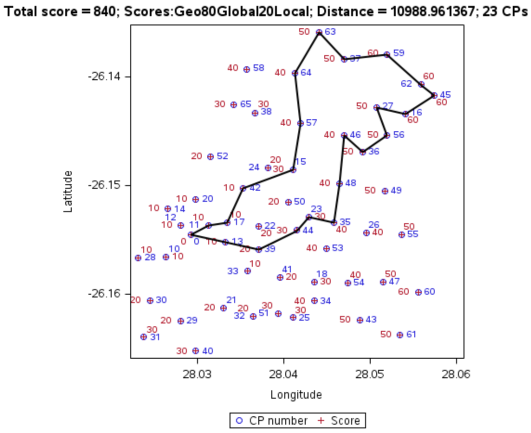
(d) 8 km



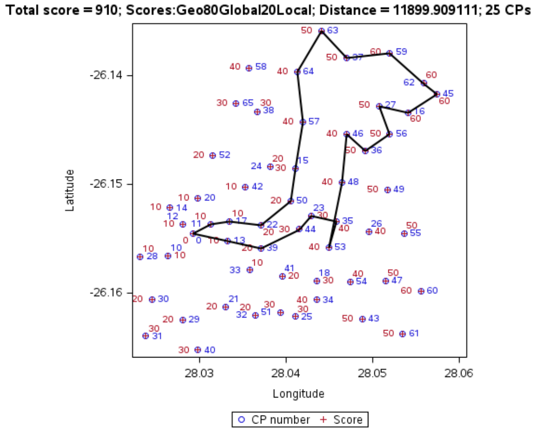
(e) 9 km



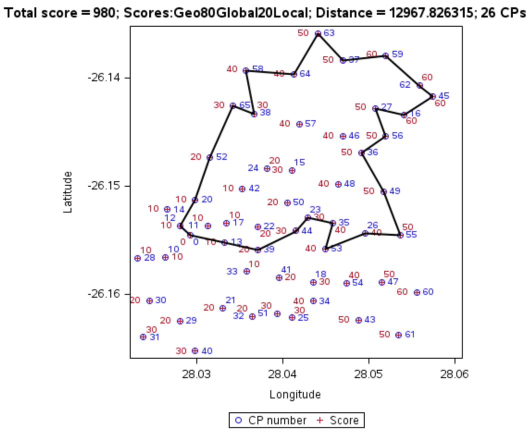
(f) 10 km



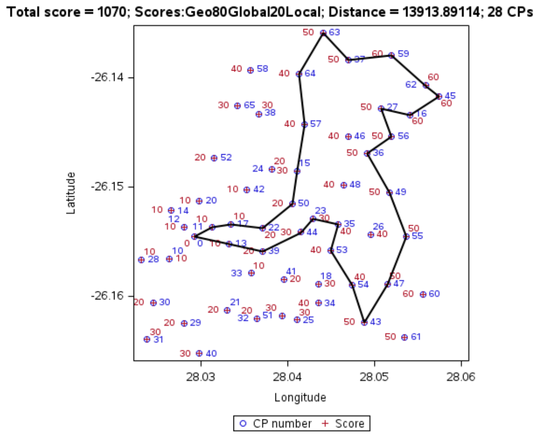
(g) 11 km



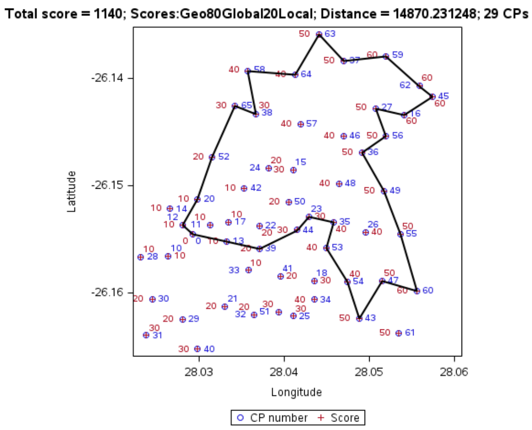
(h) 12 km



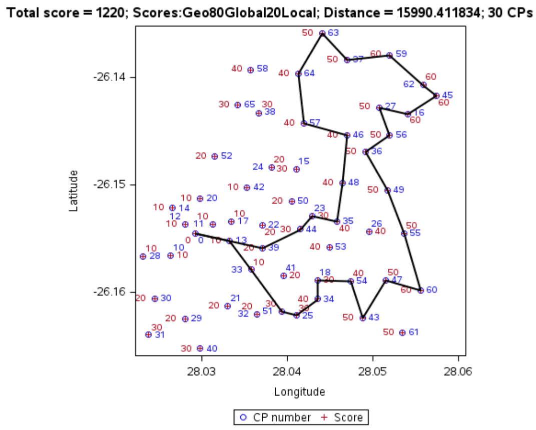
(i) 13 km



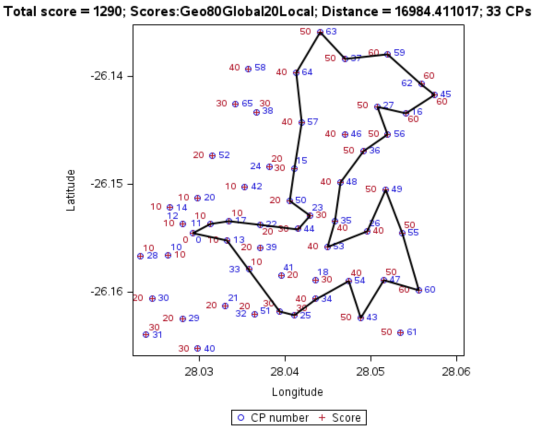
(j) 14 km



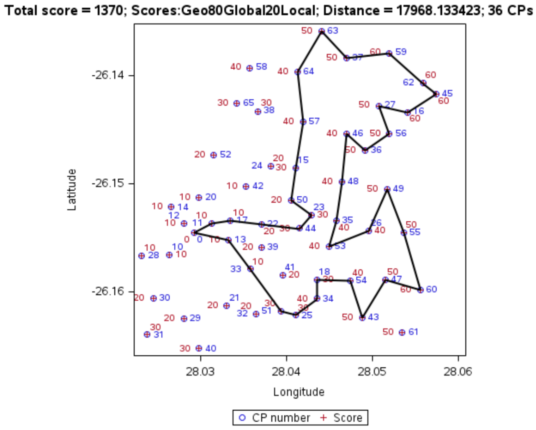
(k) 15 km



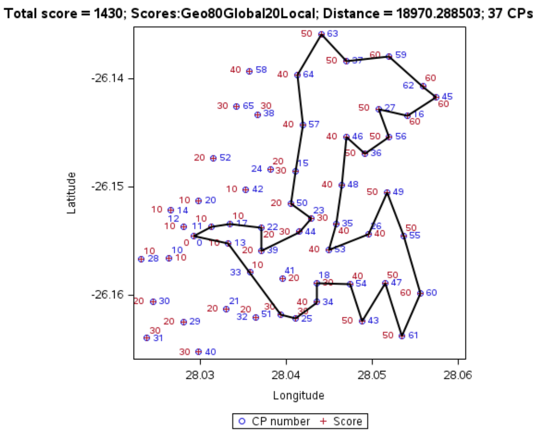
(l) 16 km



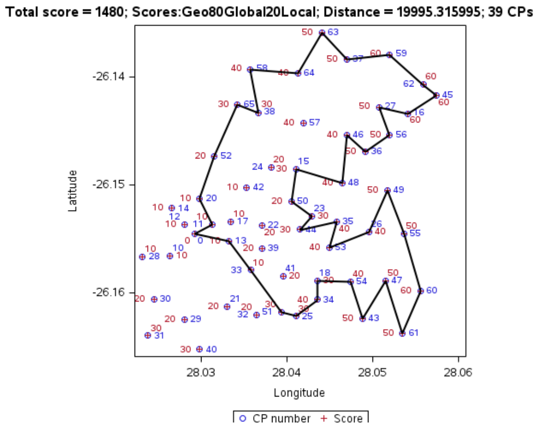
(m) 17 km



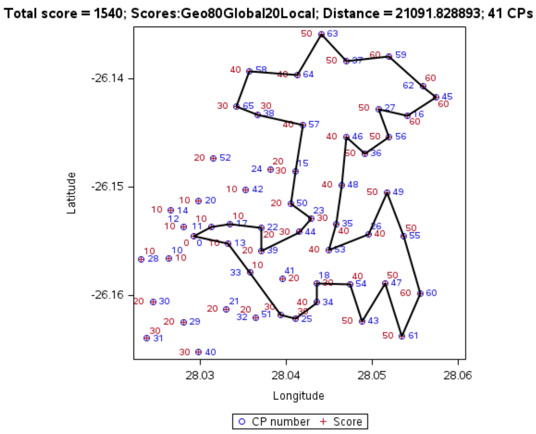
(n) 18 km



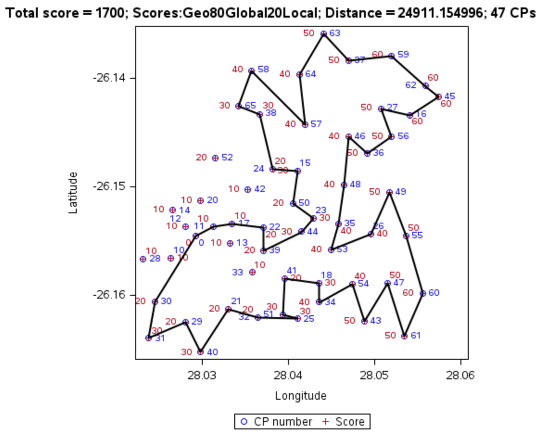
(o) 19 km



(p) 20 km

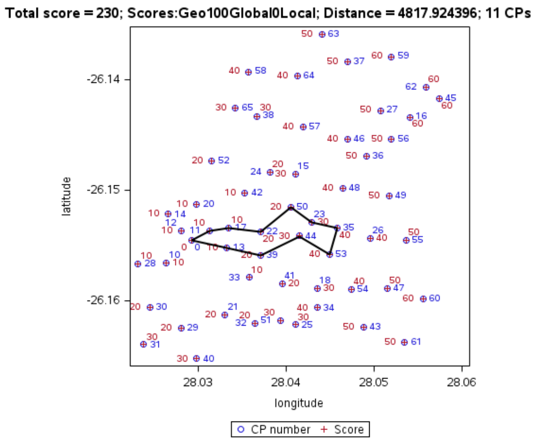


(q) 21,1 km

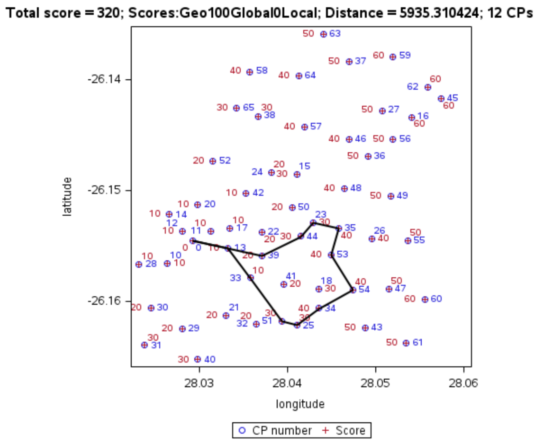


(r) 25 km

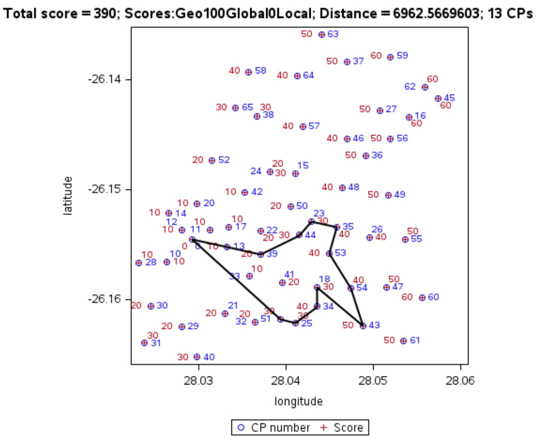
K.12 Geo100G0L OP output



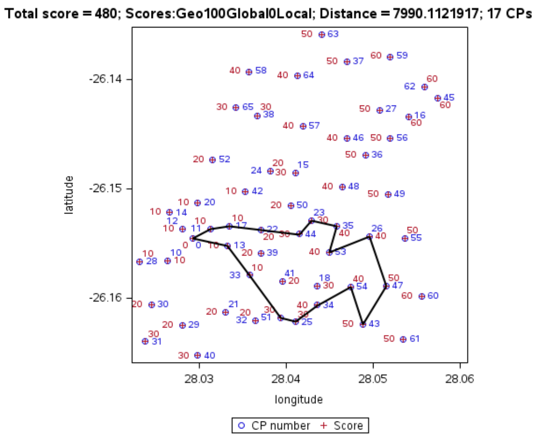
(a) 5 km



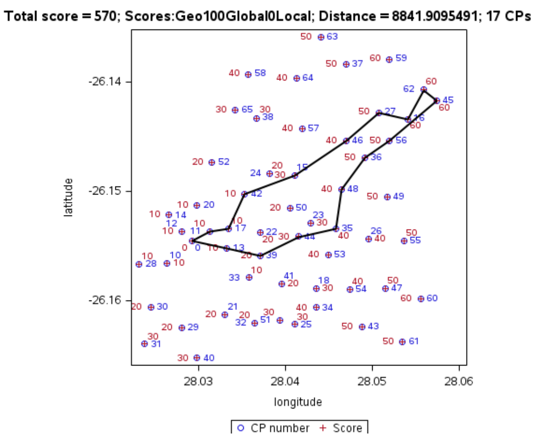
(b) 6 km



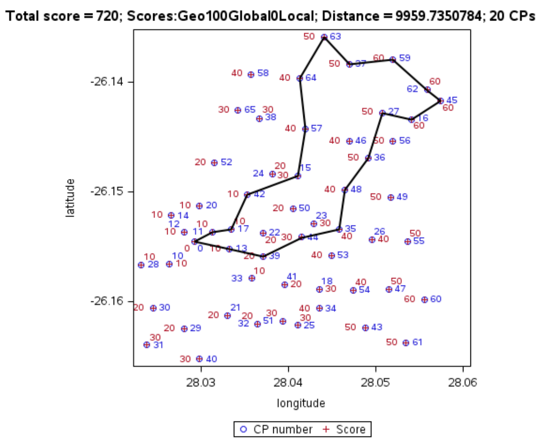
(c) 7 km



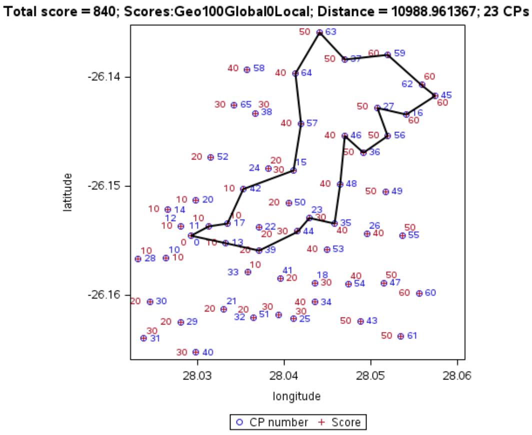
(d) 8 km



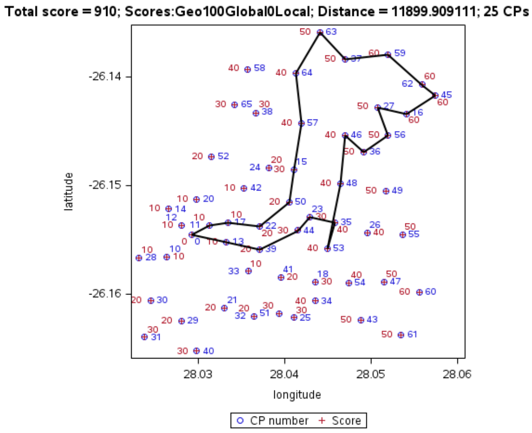
(e) 9 km



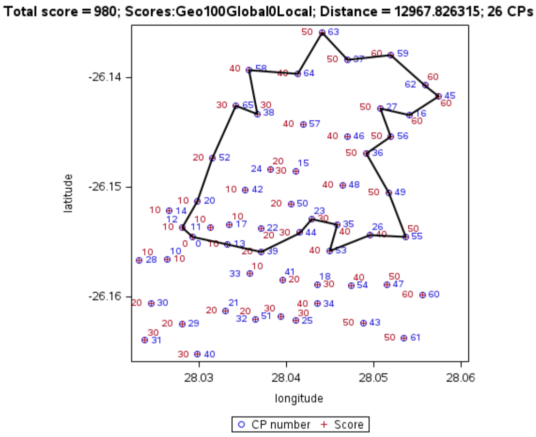
(f) 10 km



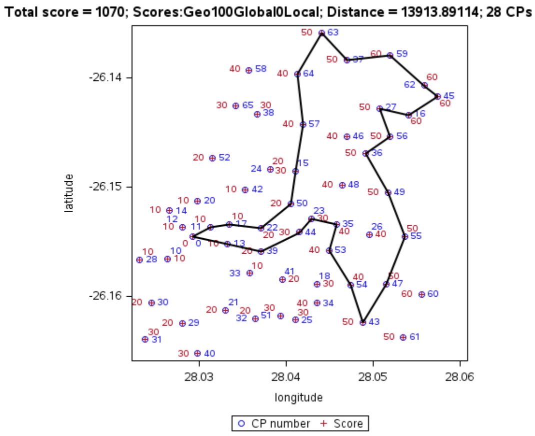
(g) 11 km



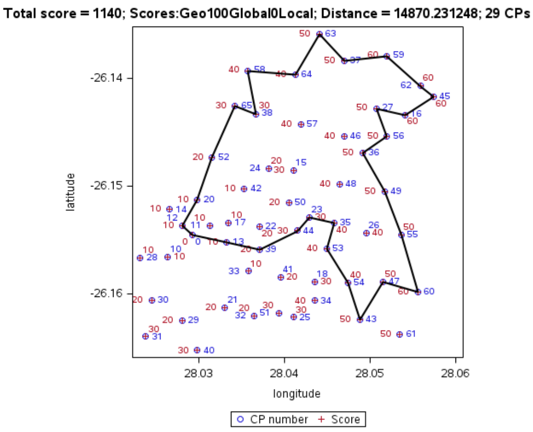
(h) 12 km



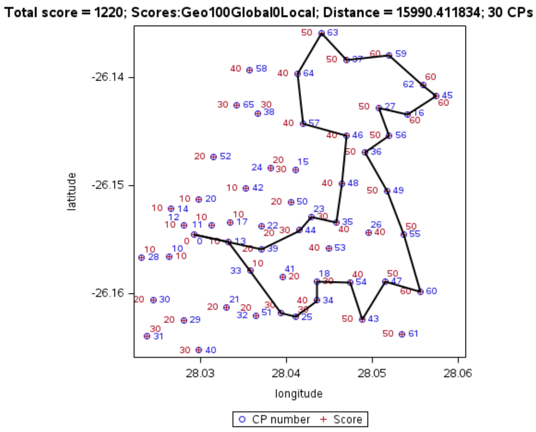
(i) 13 km



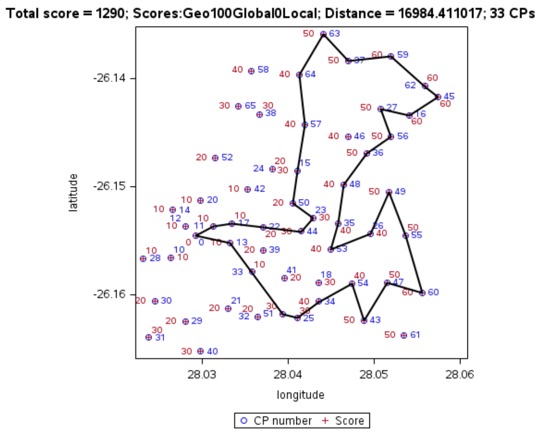
(j) 14 km



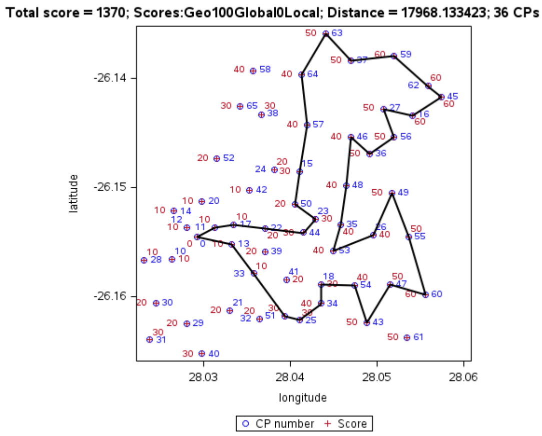
(k) 15 km



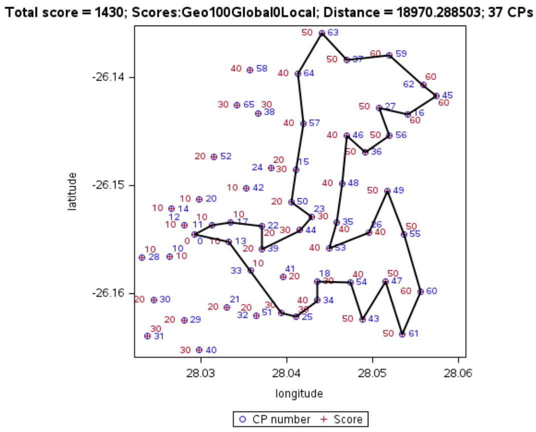
(l) 16 km



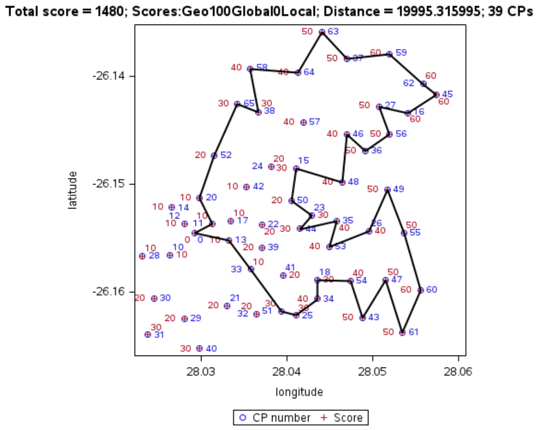
(m) 17 km



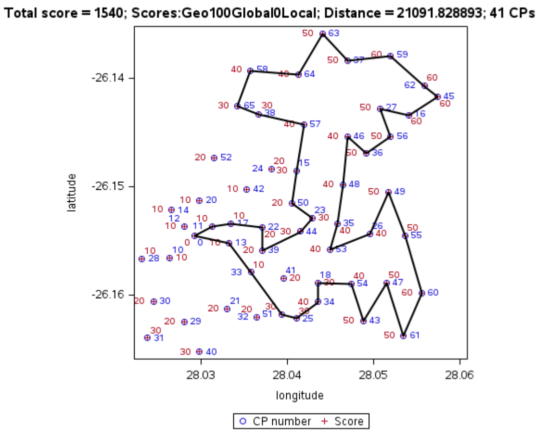
(n) 18 km



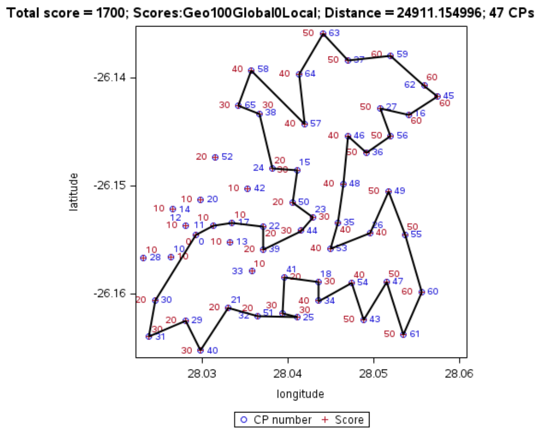
(o) 19 km



(p) 20 km

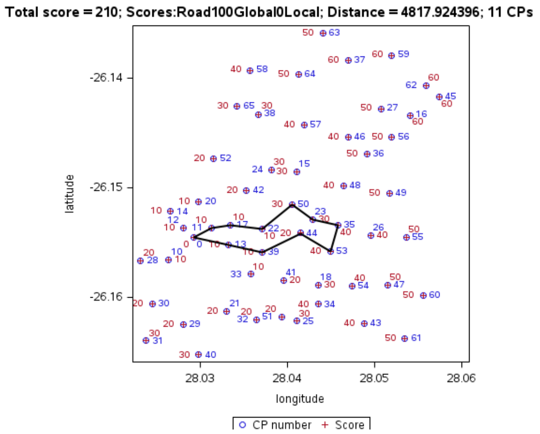


(q) 21,1 km

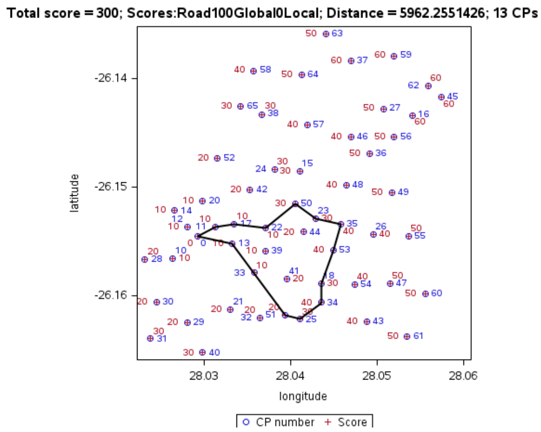


(r) 25 km

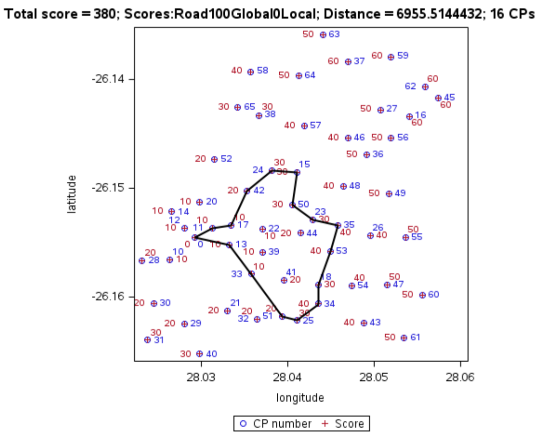
K.13 Road100G0L OP output



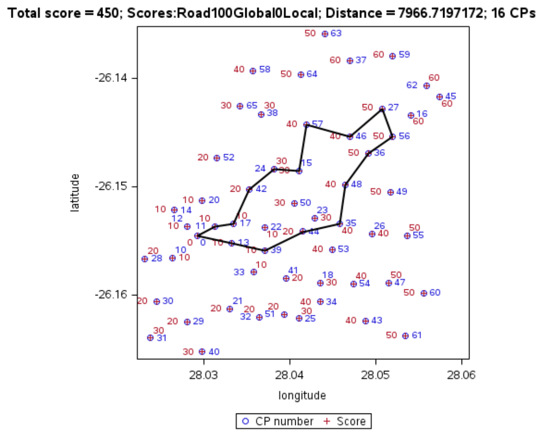
(a) 5 km



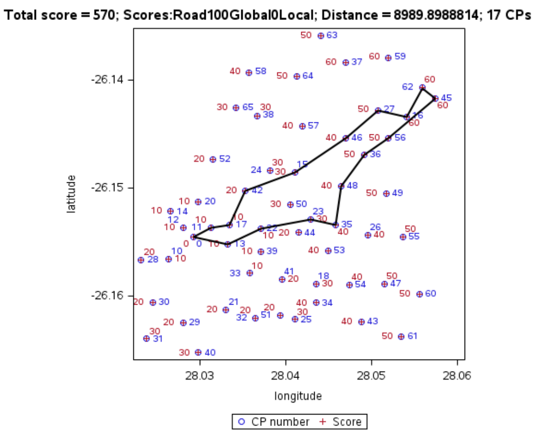
(b) 6 km



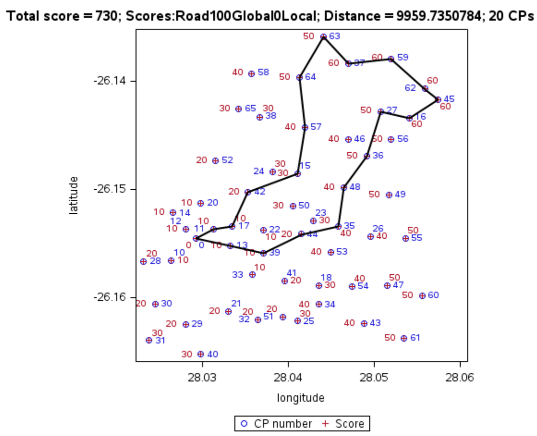
(c) 7 km



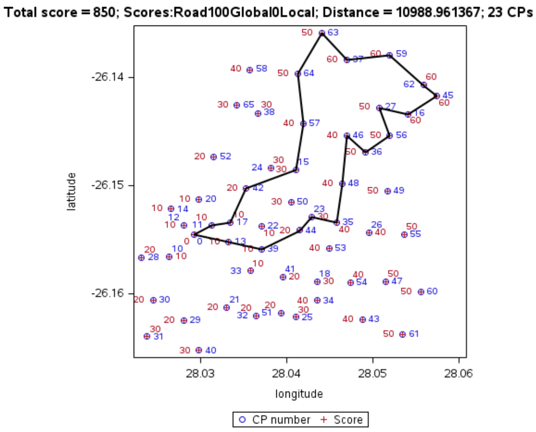
(d) 8 km



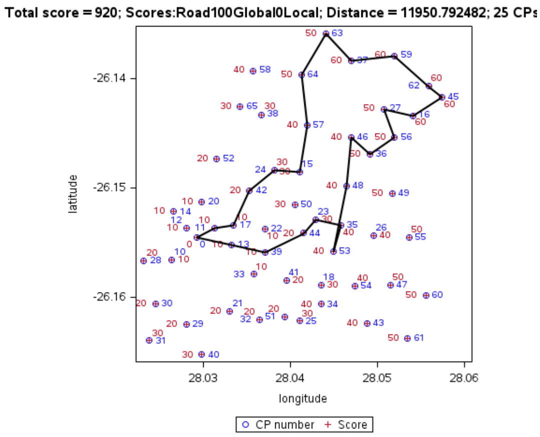
(e) 9 km



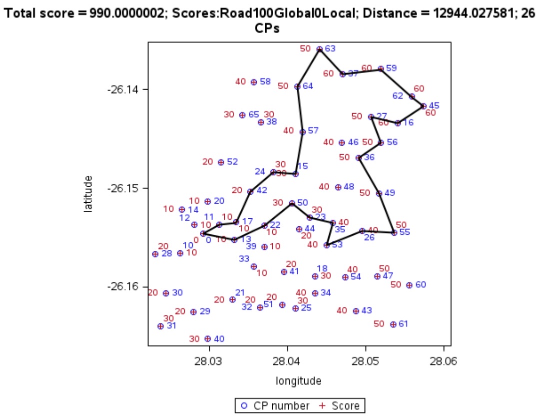
(f) 10 km



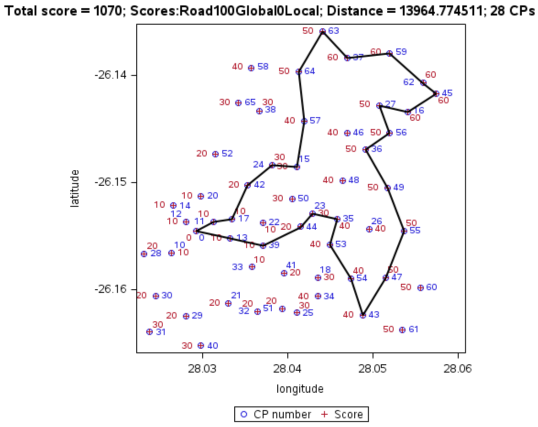
(g) 11 km



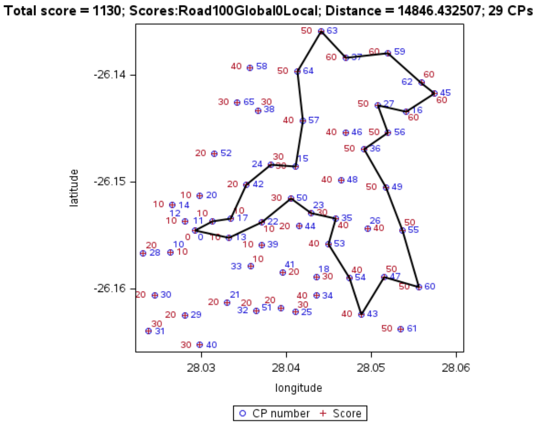
(h) 12 km



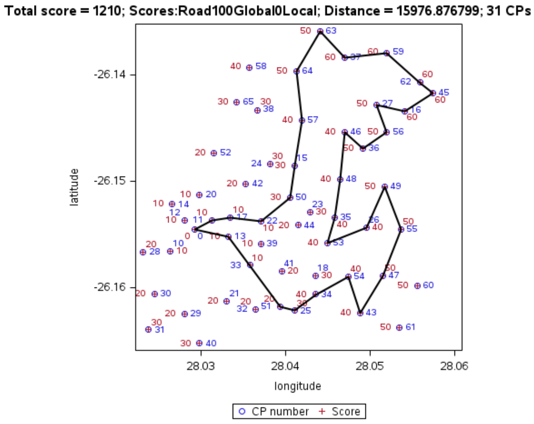
(i) 13 km



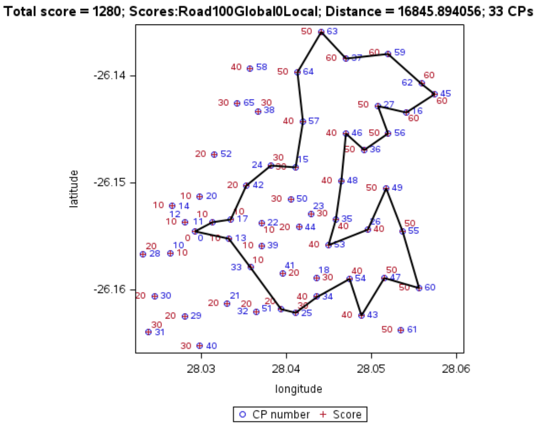
(j) 14 km



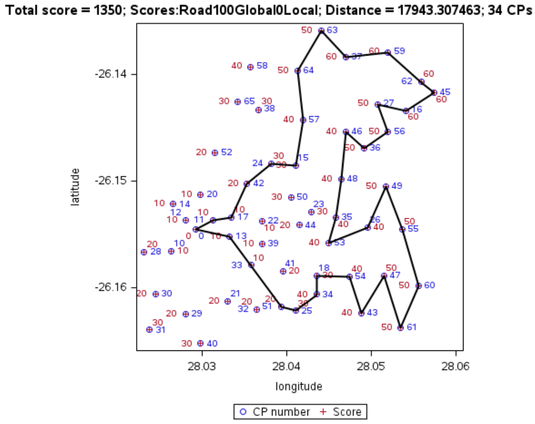
(k) 15 km



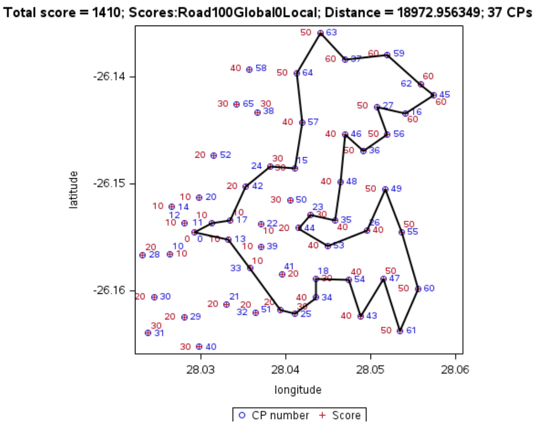
(l) 16 km



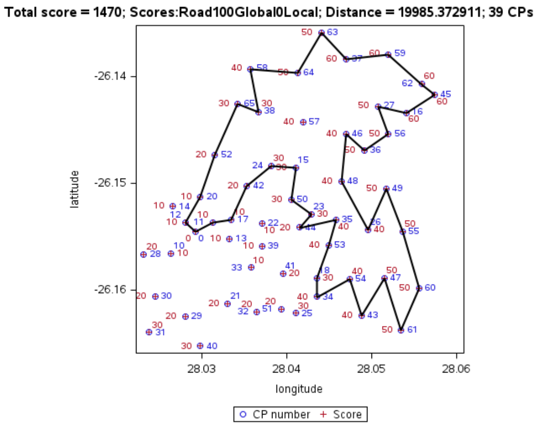
(m) 17 km



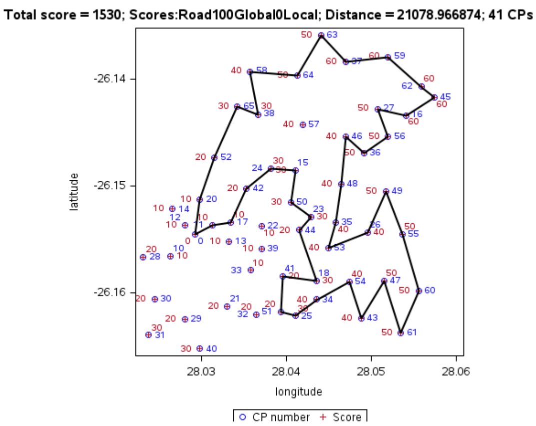
(n) 18 km



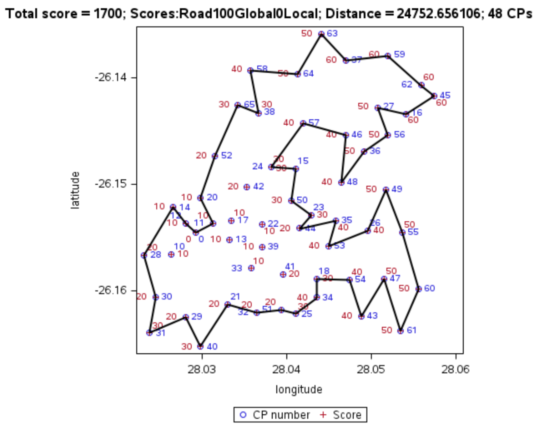
(o) 19 km



(p) 20 km

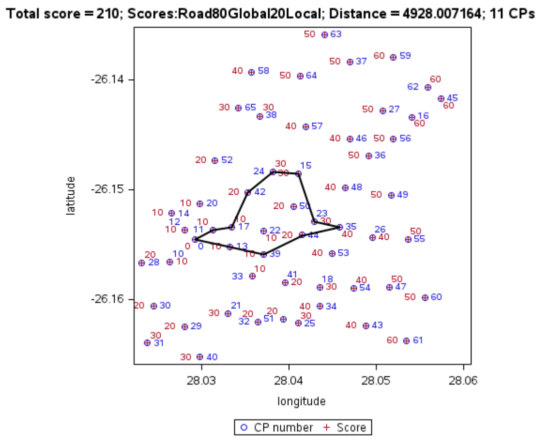


(q) 21,1 km

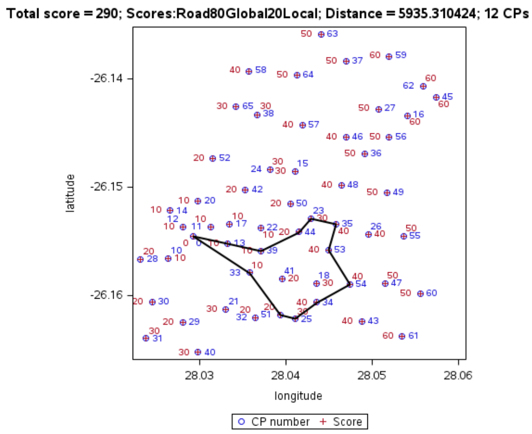


(r) 25 km

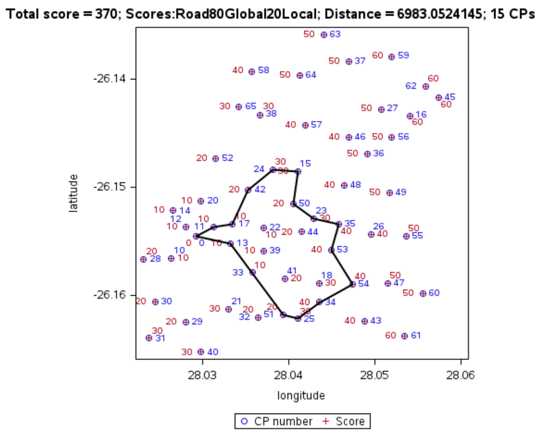
K.14 Road80G20L OP output



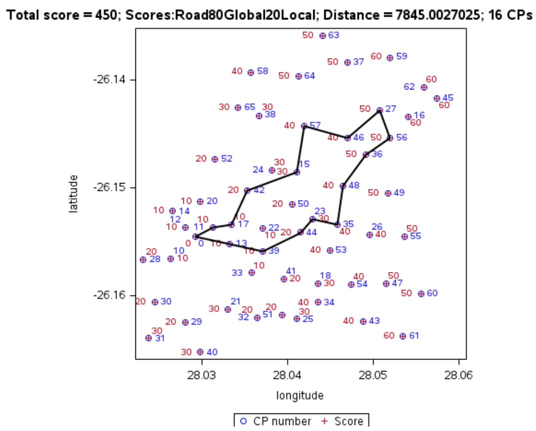
(a) 5 km



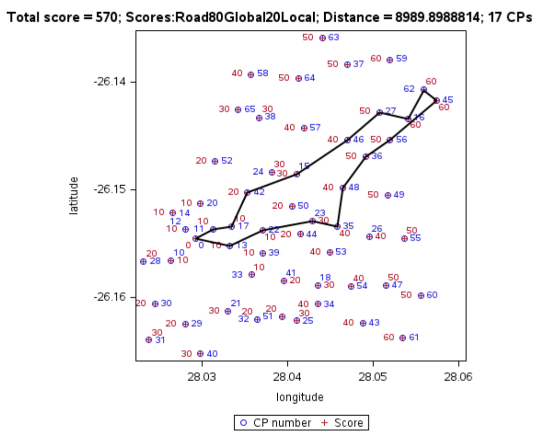
(b) 6 km



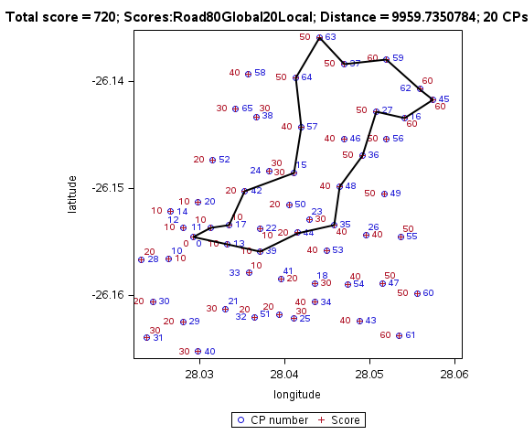
(c) 7 km



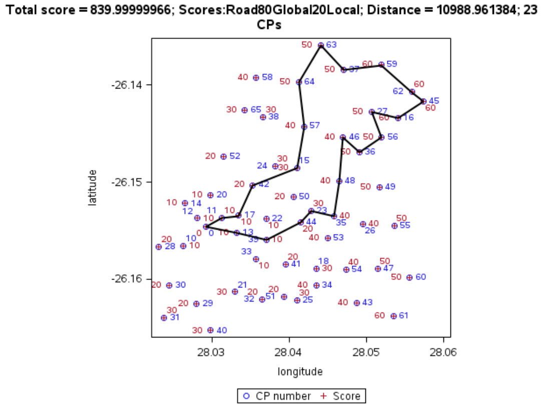
(d) 8 km



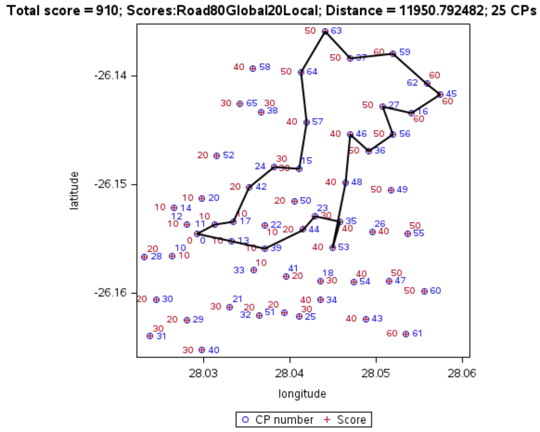
(e) 9 km



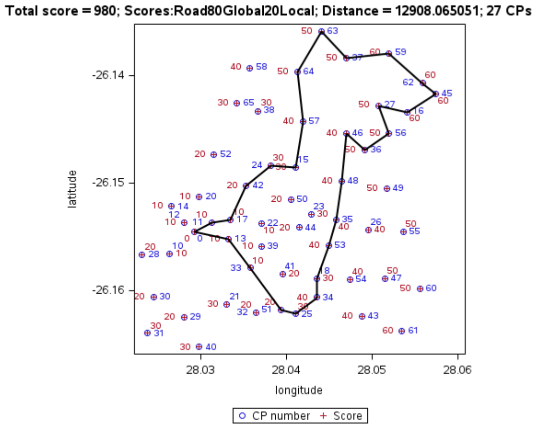
(f) 10 km



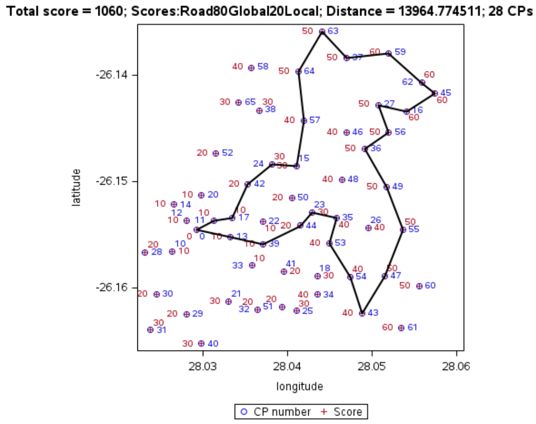
(g) 11 km



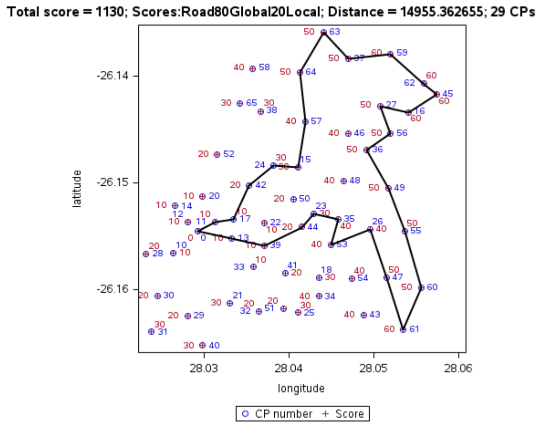
(h) 12 km



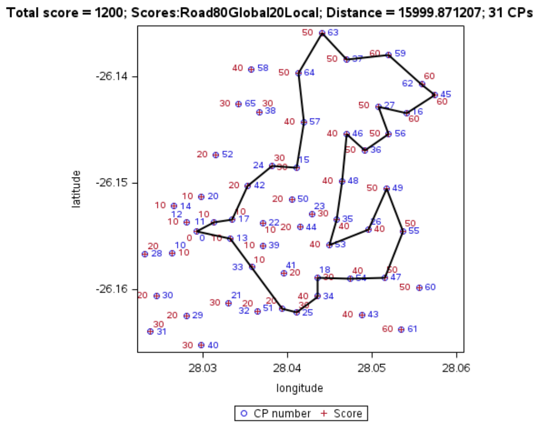
(i) 13 km



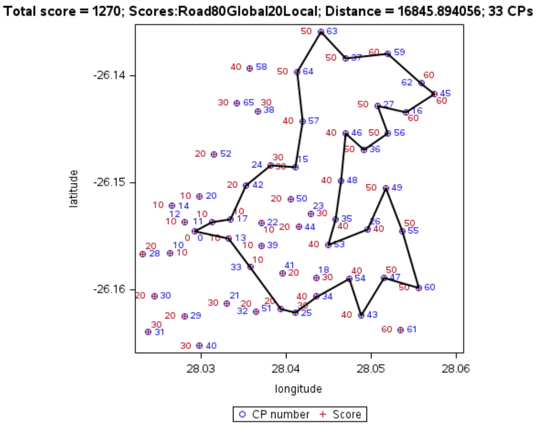
(j) 14 km



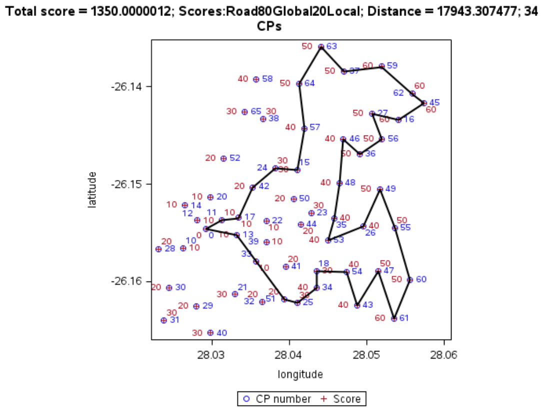
(k) 15 km



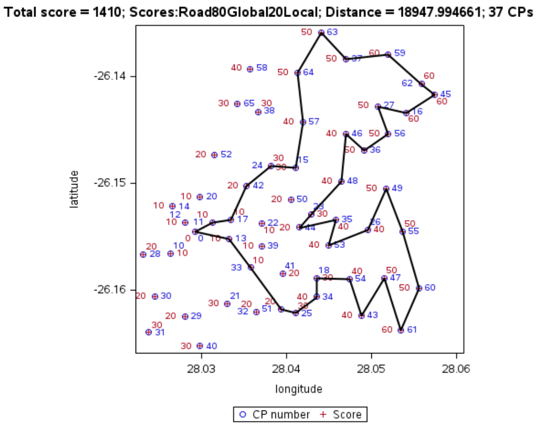
(l) 16 km



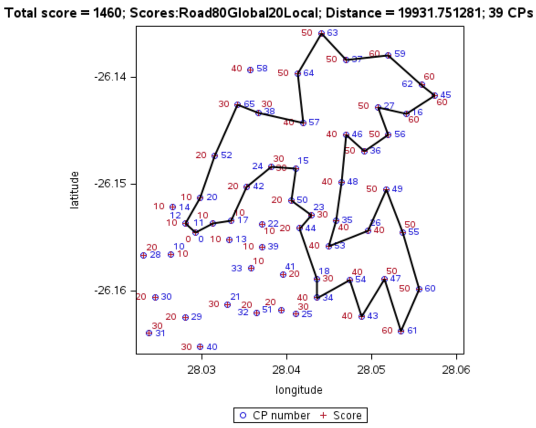
(m) 17 km



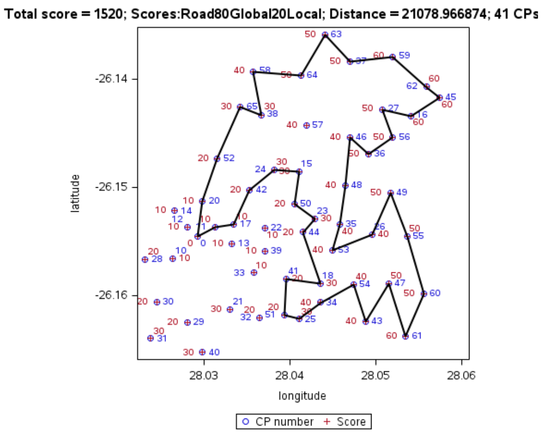
(n) 18 km



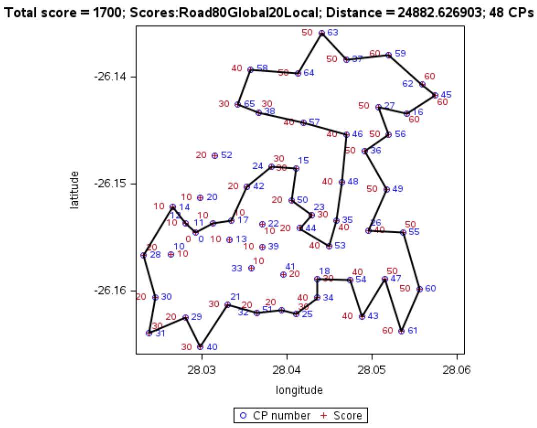
(o) 19 km



(p) 20 km

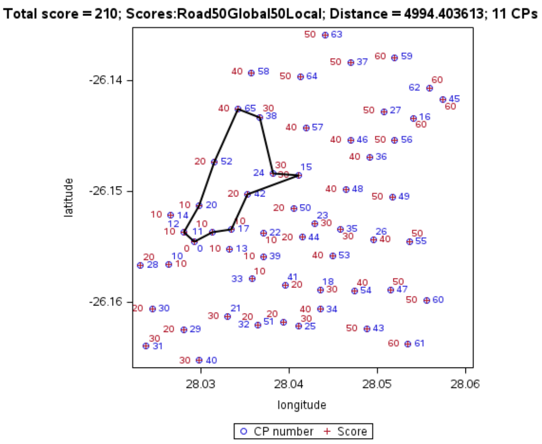


(q) 21,1 km

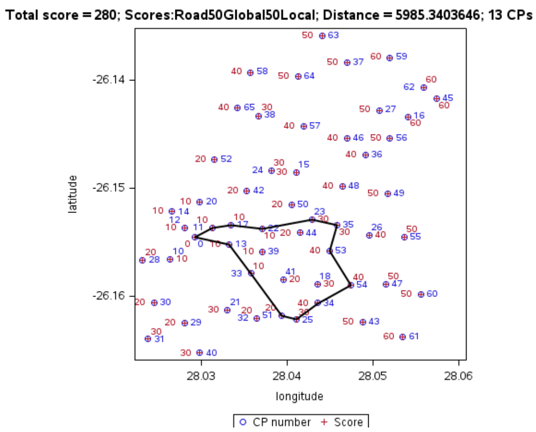


(r) 25 km

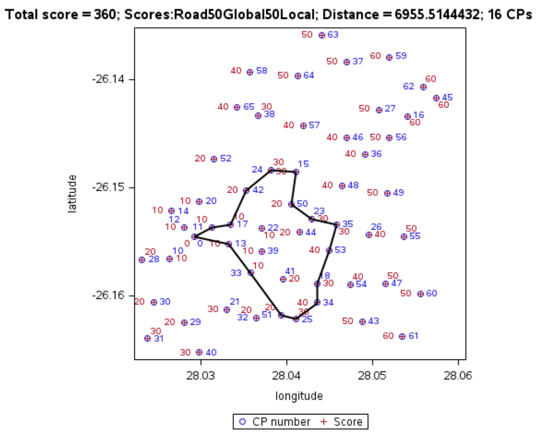
K.15 Road50G50L OP output



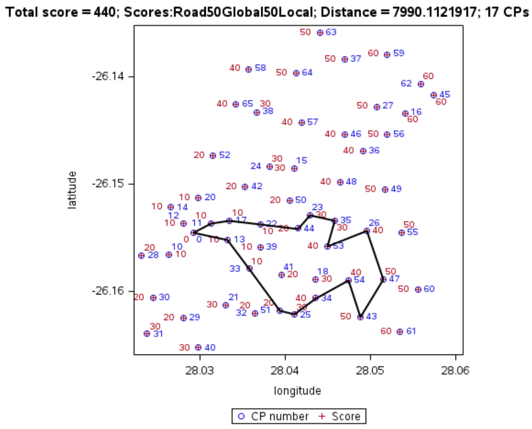
(a) 5 km



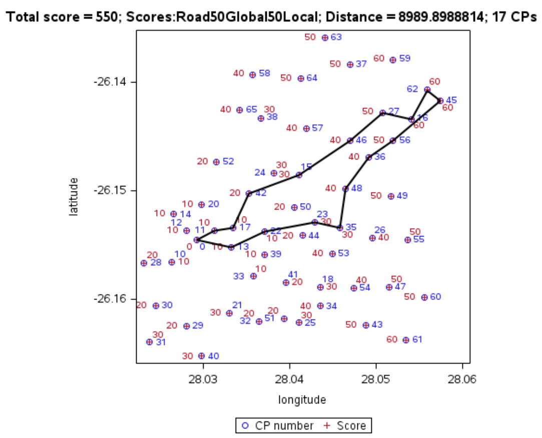
(b) 6 km



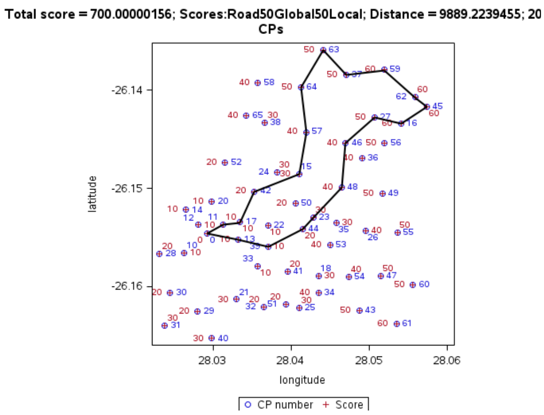
(c) 7 km



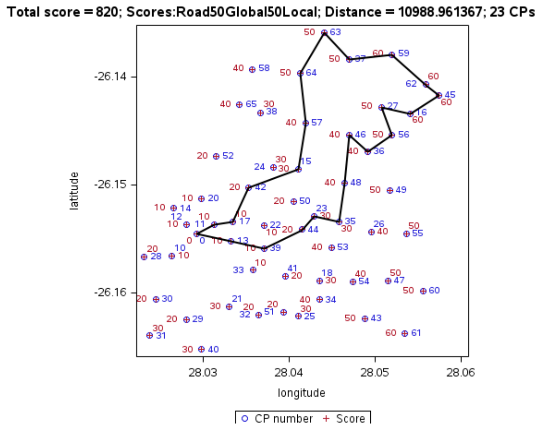
(d) 8 km



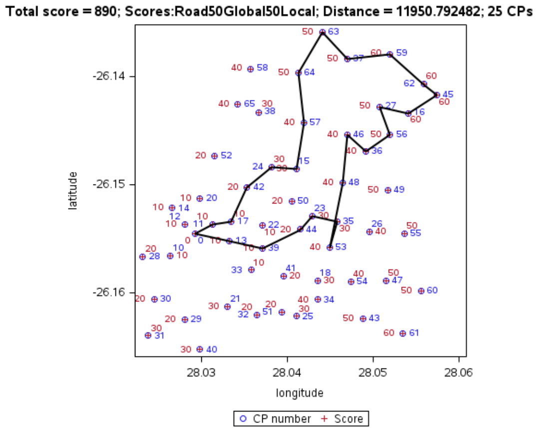
(e) 9 km



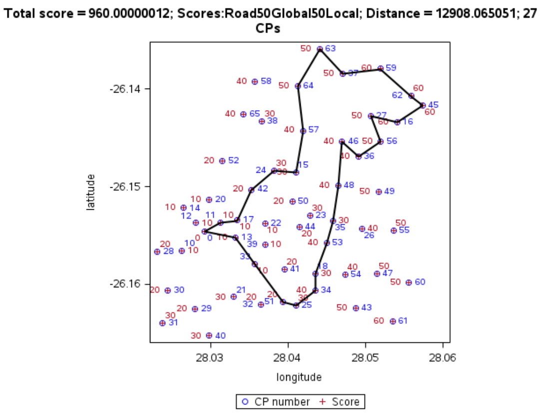
(f) 10 km



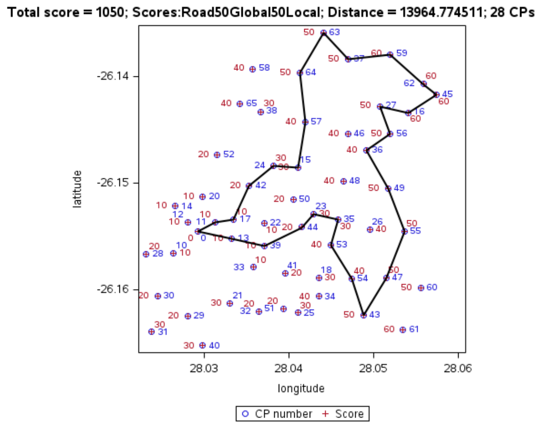
(g) 11 km



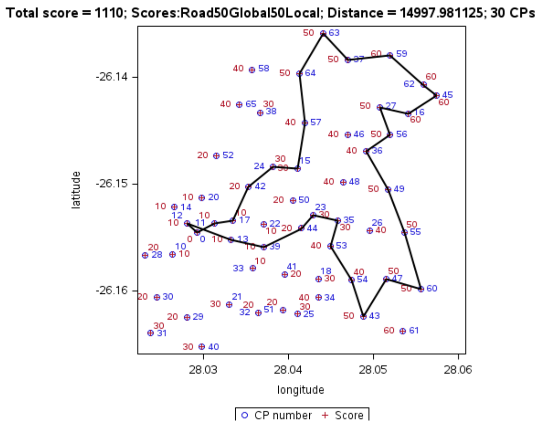
(h) 12 km



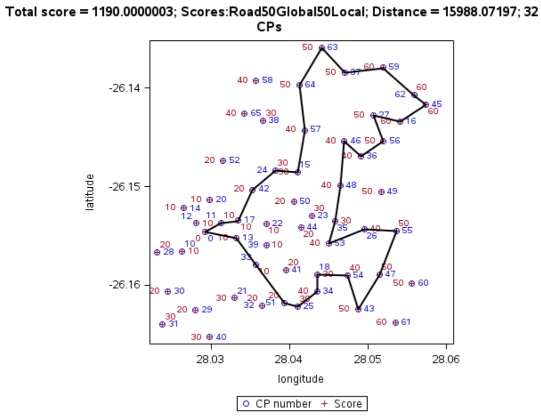
(i) 13 km



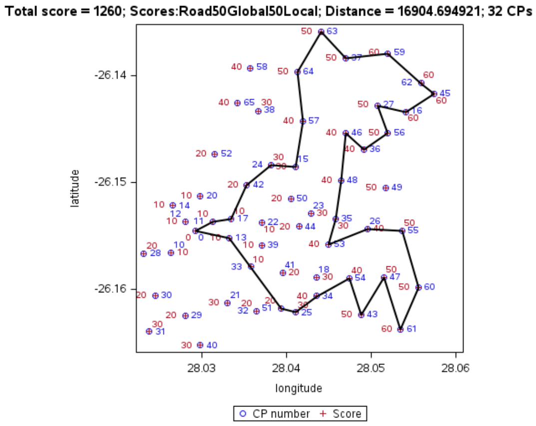
(j) 14 km



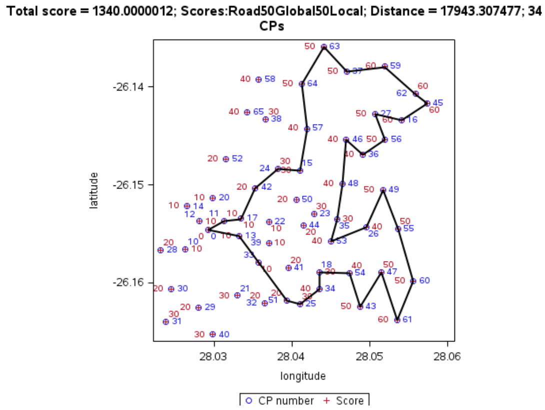
(k) 15 km



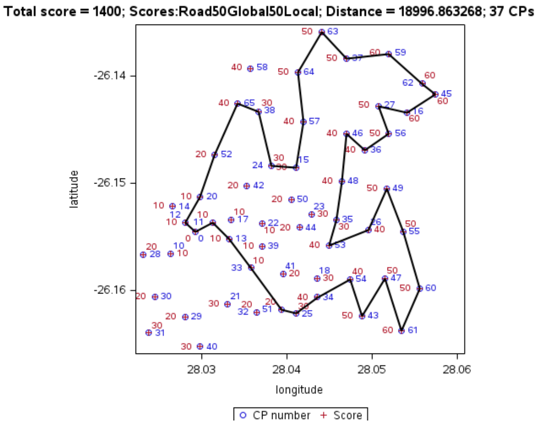
(l) 16 km



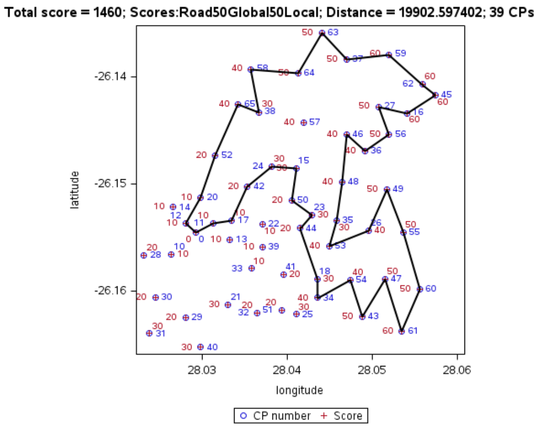
(m) 17 km



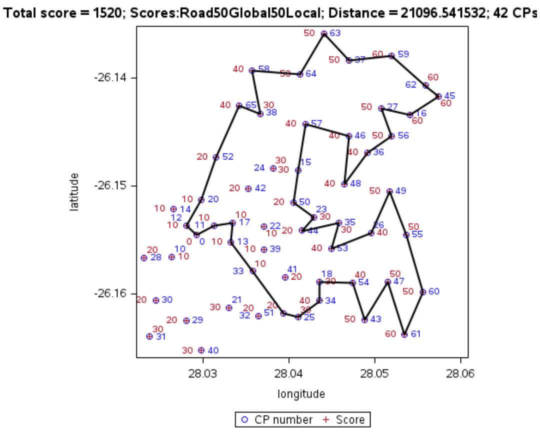
(n) 18 km



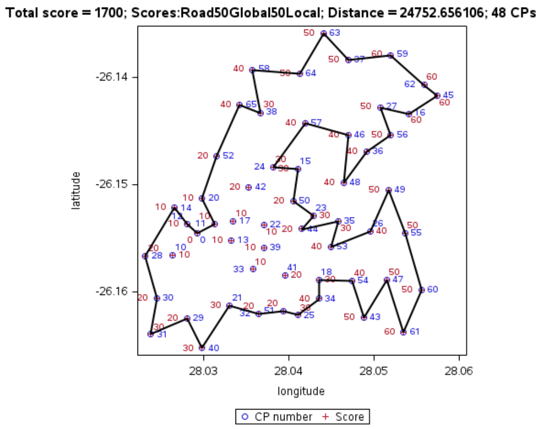
(o) 19 km



(p) 20 km

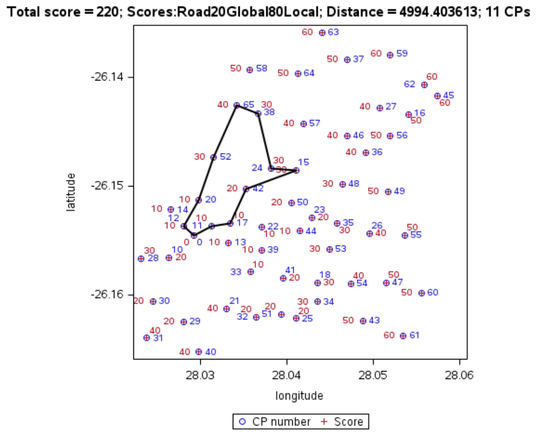


(q) 21,1 km

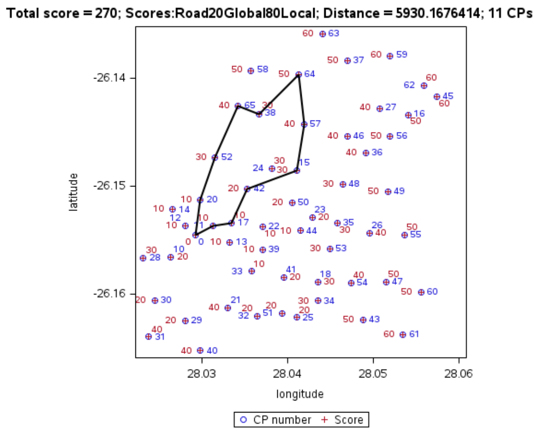


(r) 25 km

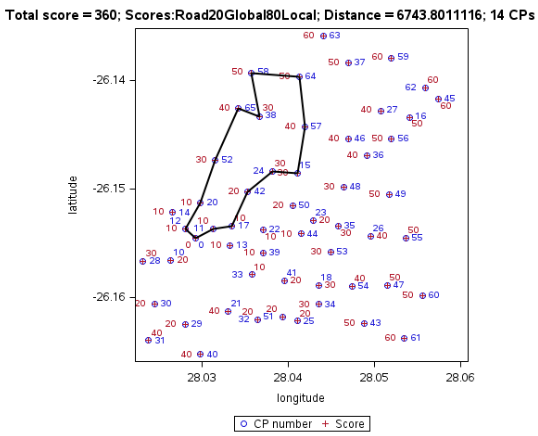
K.16 Road20G80L OP output



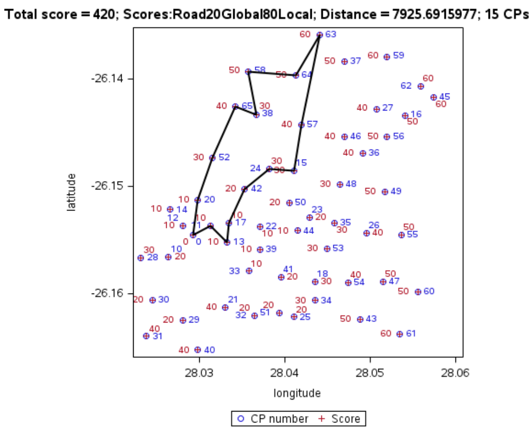
(a) 5 km



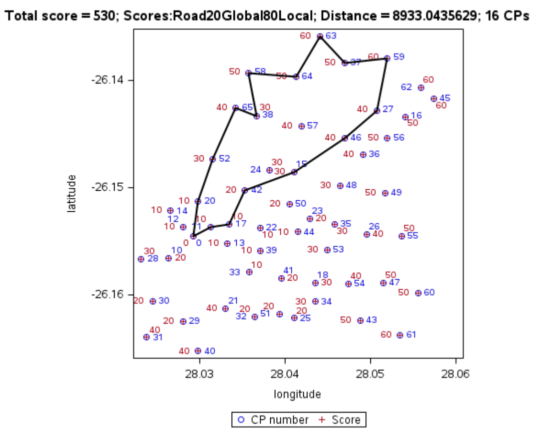
(b) 6 km



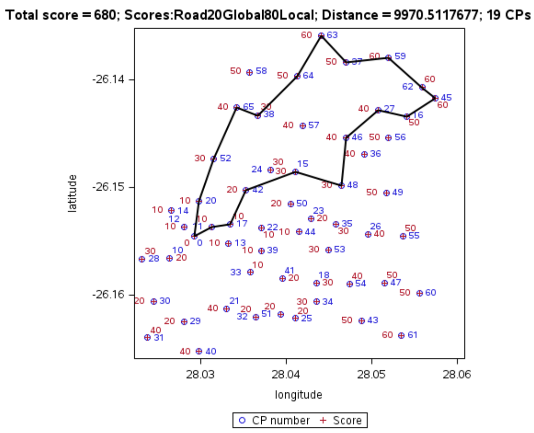
(c) 7 km



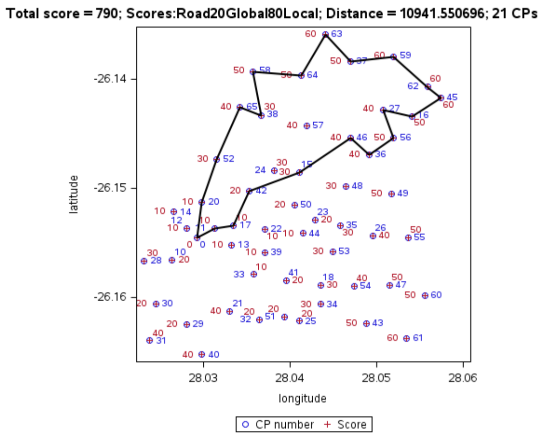
(d) 8 km



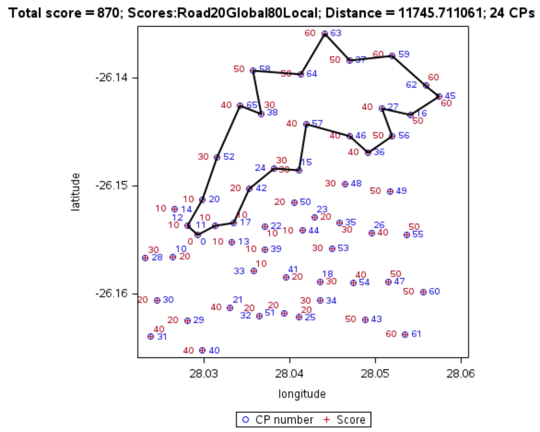
(e) 9 km



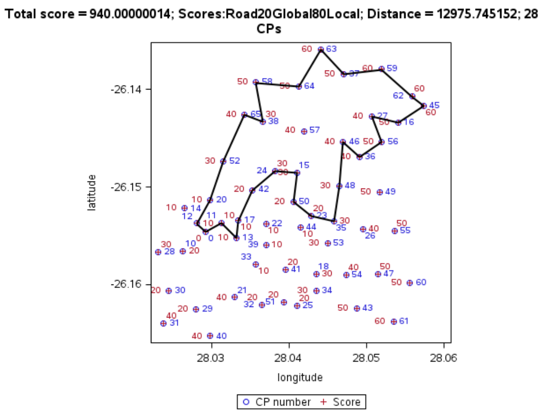
(f) 10 km



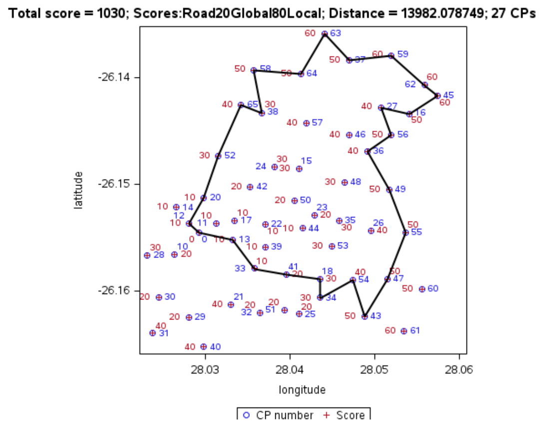
(g) 11 km



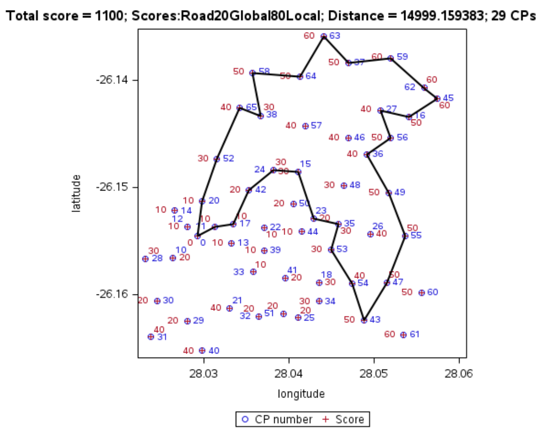
(h) 12 km



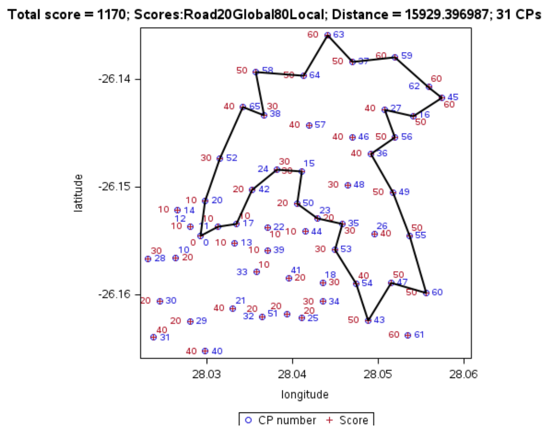
(i) 13 km



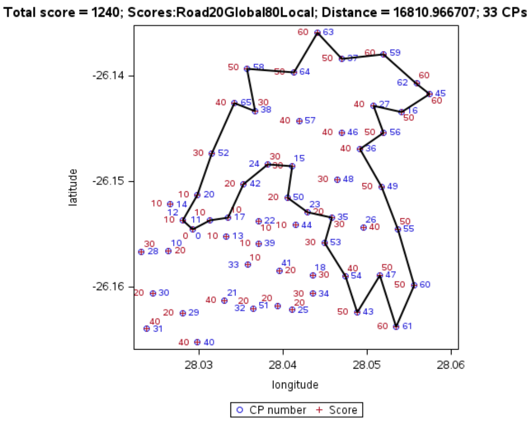
(j) 14 km



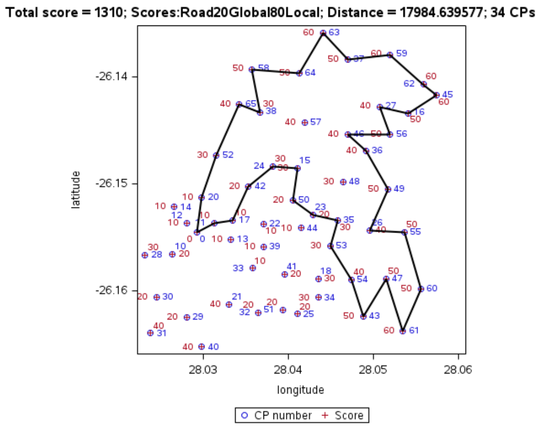
(k) 15 km



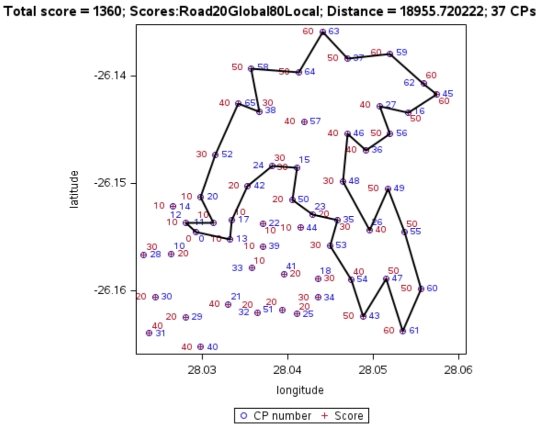
(l) 16 km



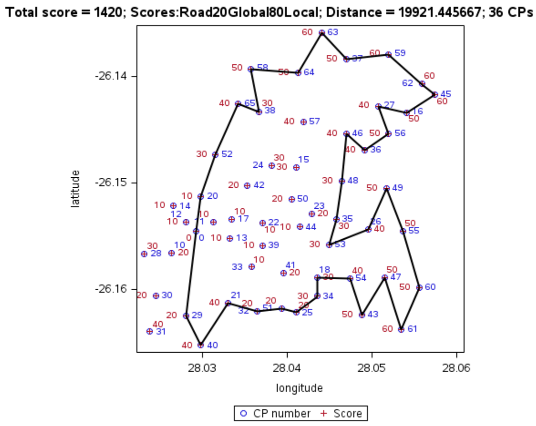
(m) 17 km



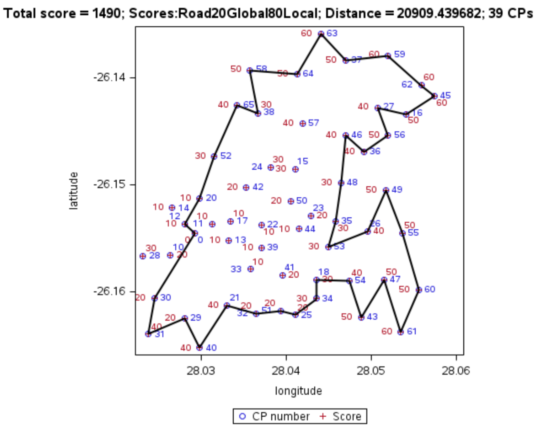
(n) 18 km



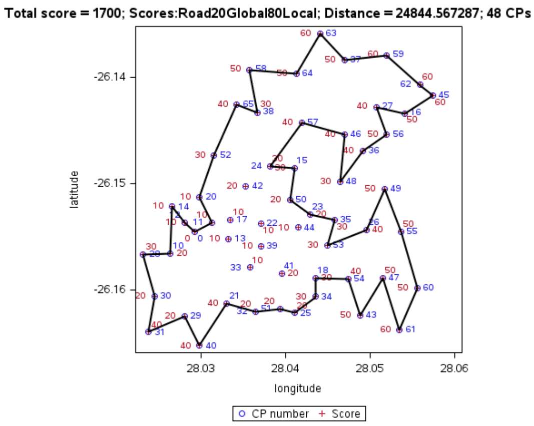
(o) 19 km



(p) 20 km

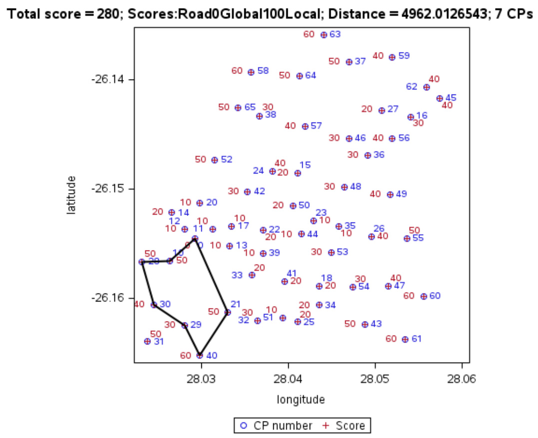


(q) 21,1 km

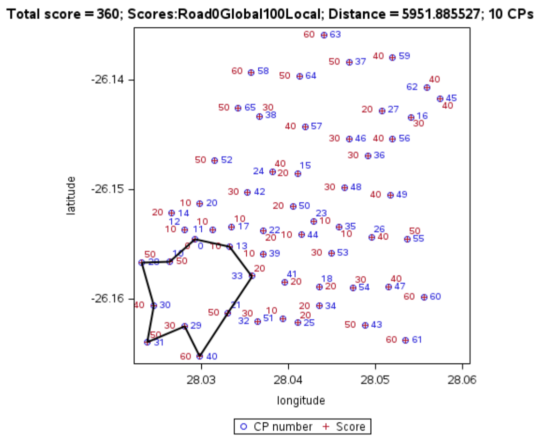


(r) 25 km

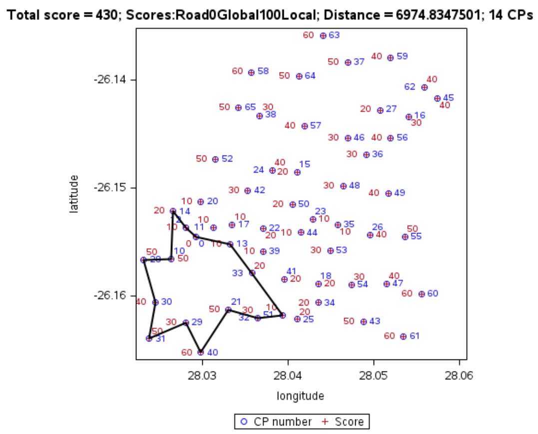
K.17 Road0G100L OP output



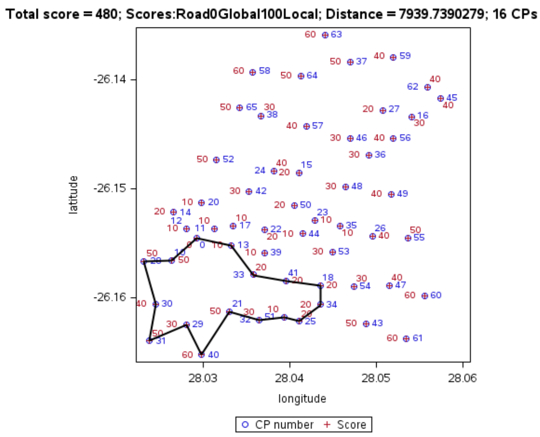
(a) 5 km



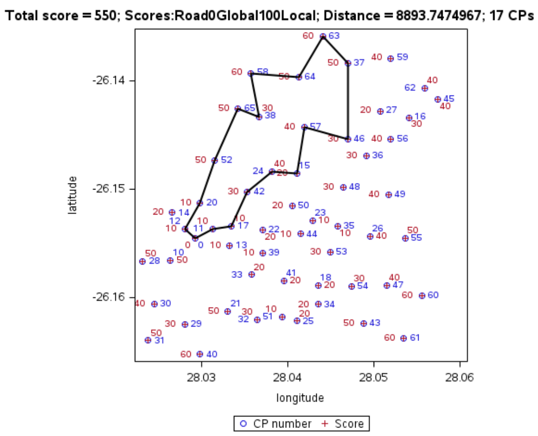
(b) 6 km



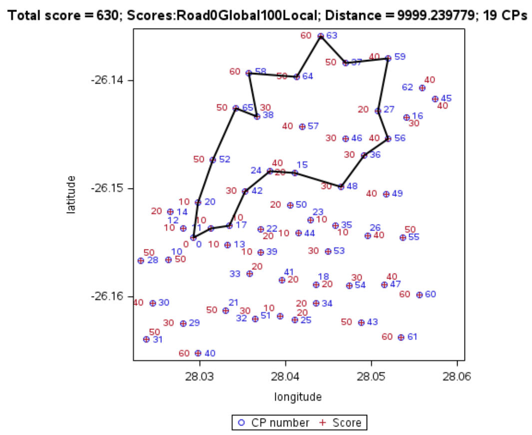
(c) 7 km



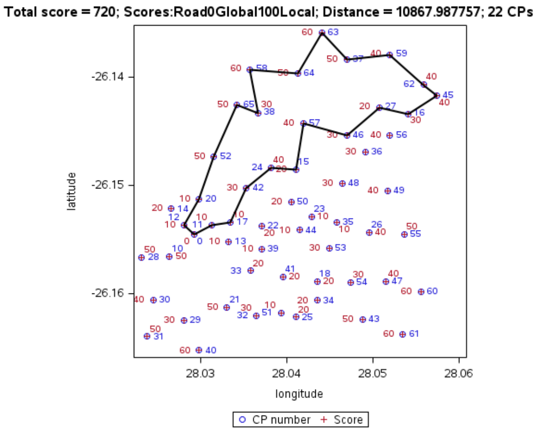
(d) 8 km



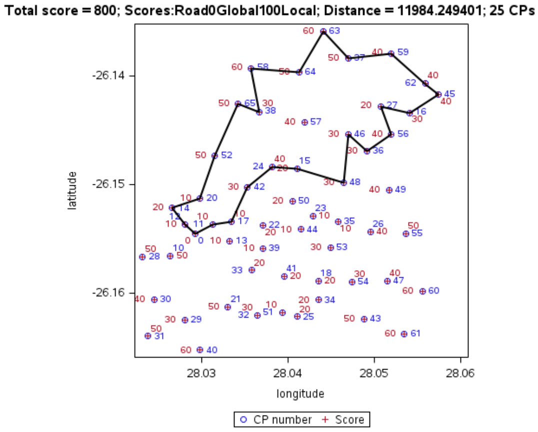
(e) 9 km



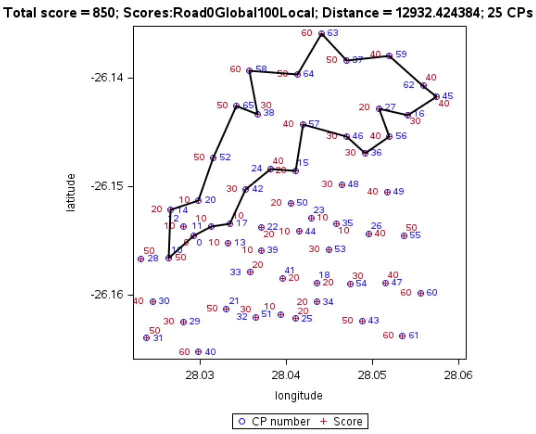
(f) 10 km



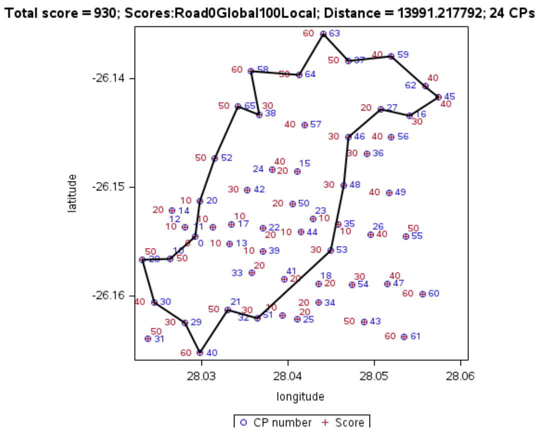
(g) 11 km



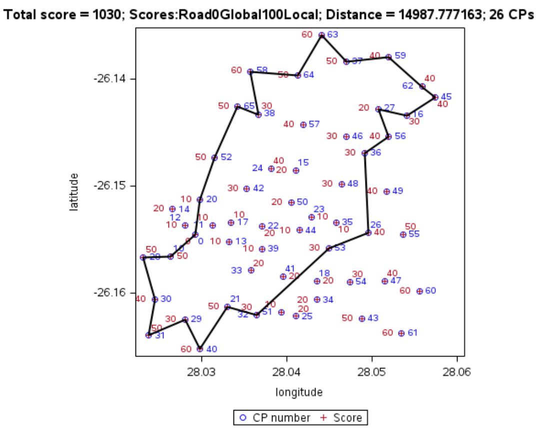
(h) 12 km



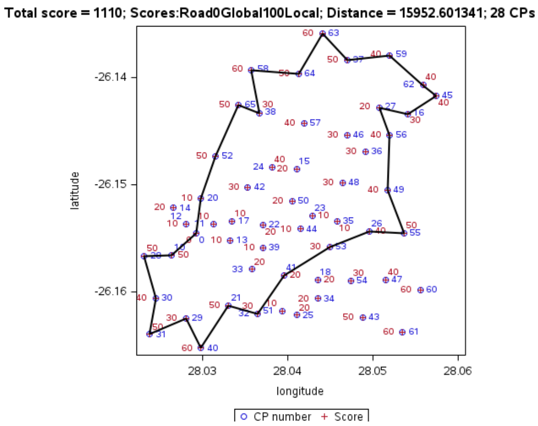
(i) 13 km



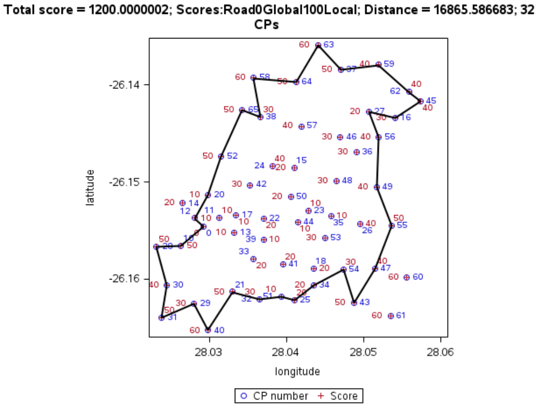
(j) 14 km



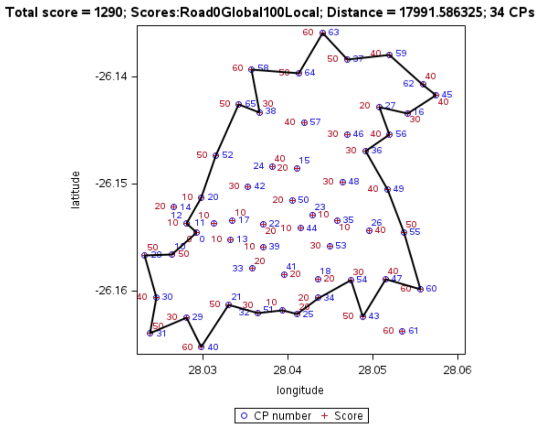
(k) 15 km



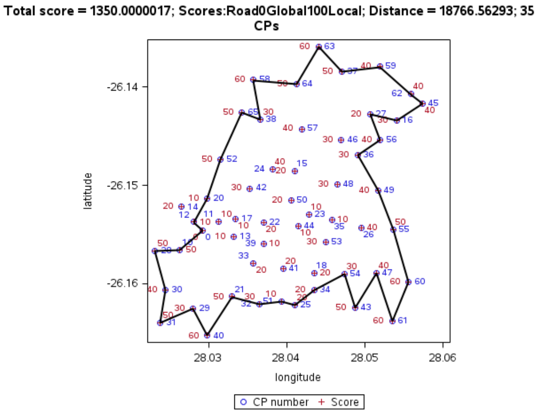
(l) 16 km



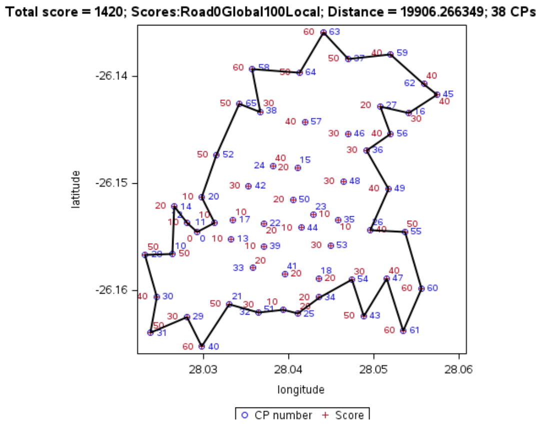
(m) 17 km



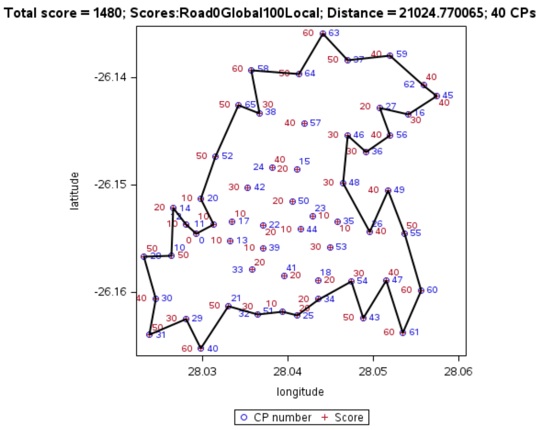
(n) 18 km



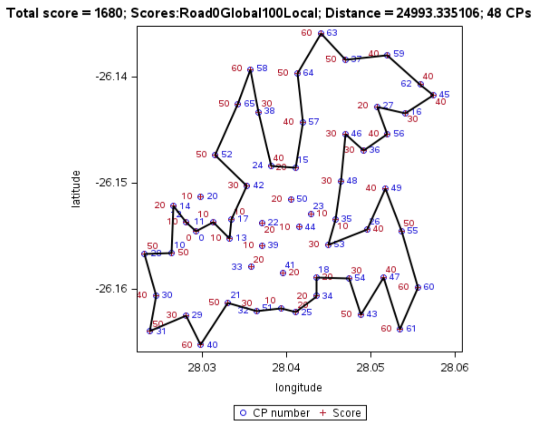
(o) 19 km



(p) 20 km

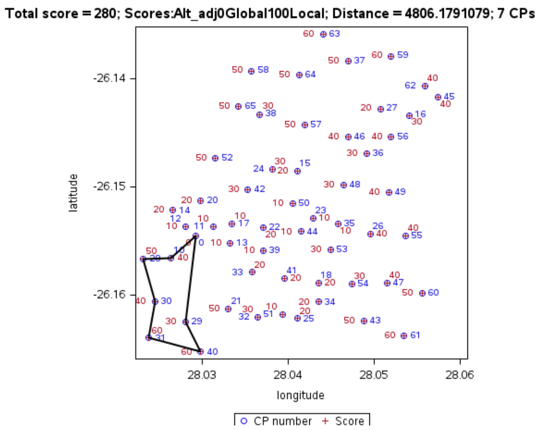


(q) 21,1 km

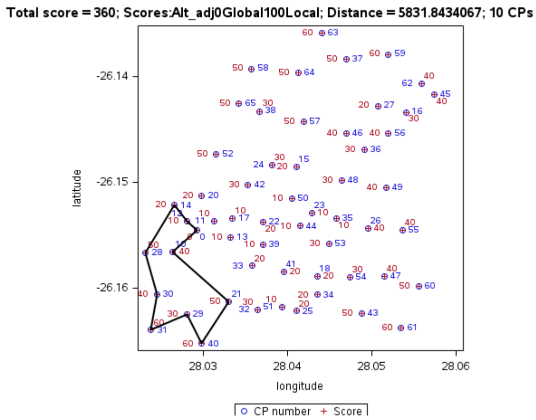


(r) 25 km

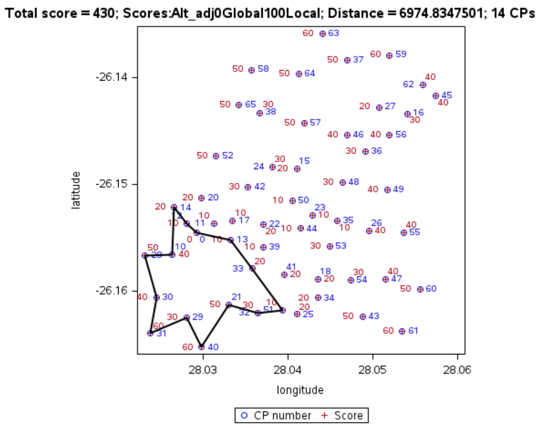
K.18 Alt_adj0G100L OP output



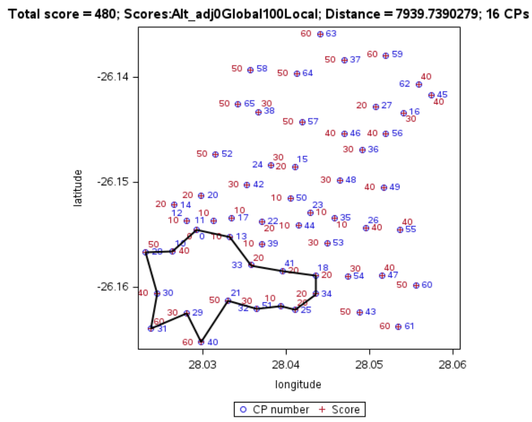
(a) 5 km



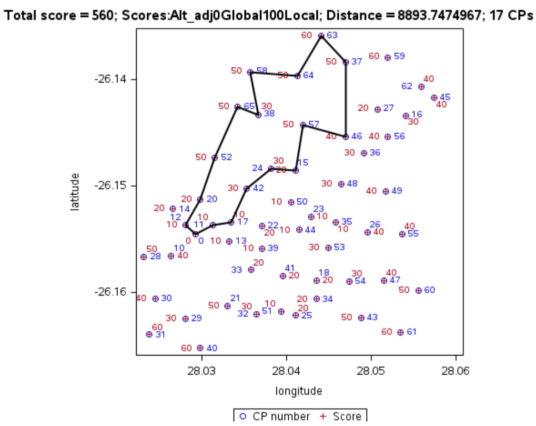
(b) 6 km



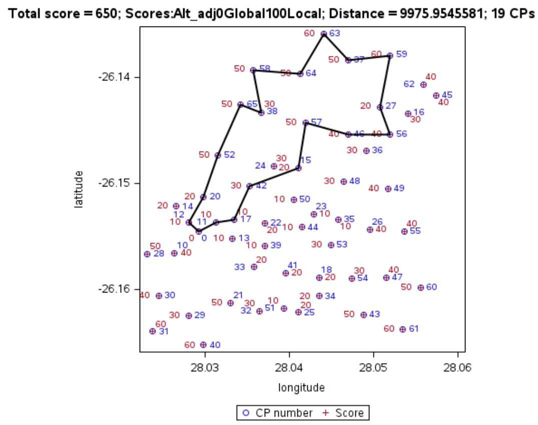
(c) 7 km



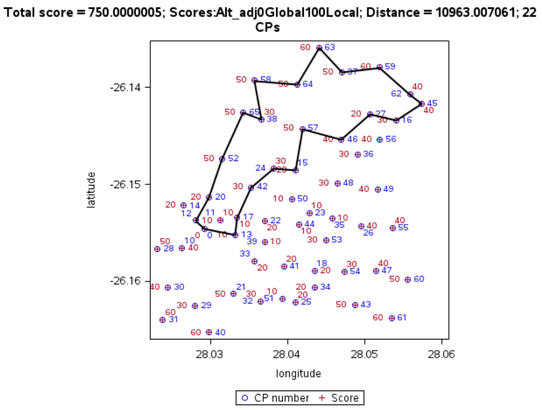
(d) 8 km



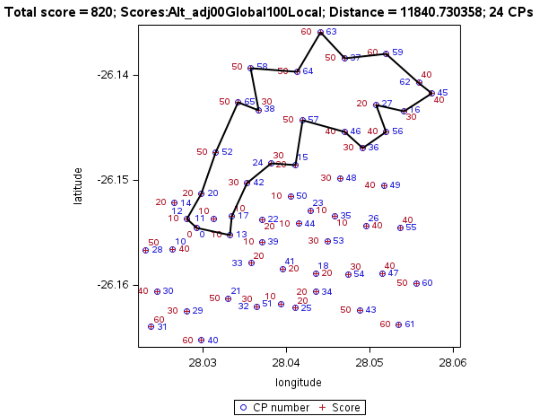
(e) 9 km



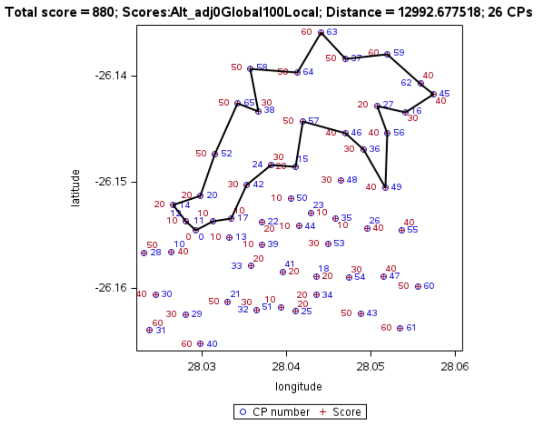
(f) 10 km



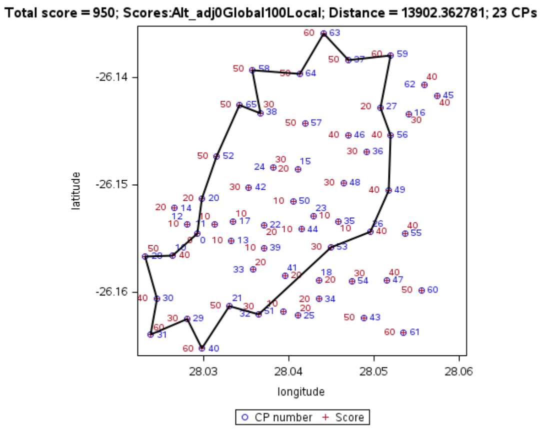
(g) 11 km



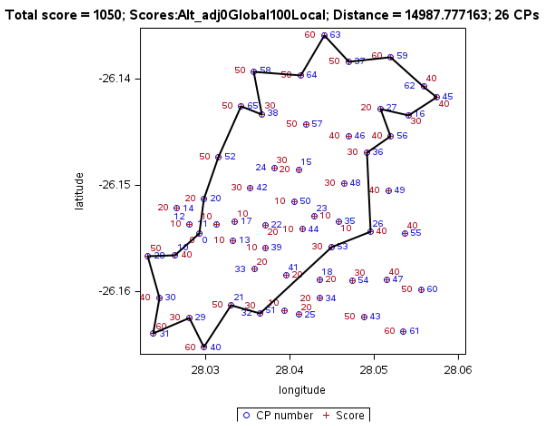
(h) 12 km



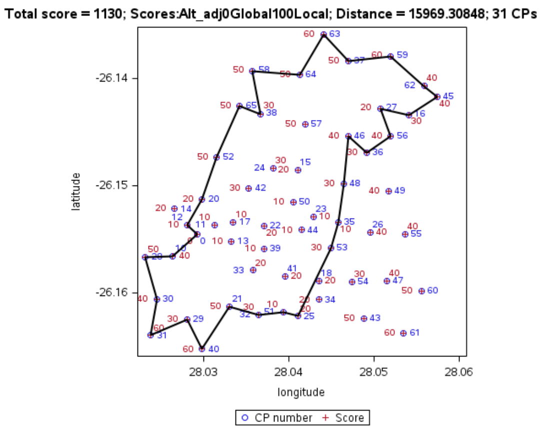
(i) 13 km



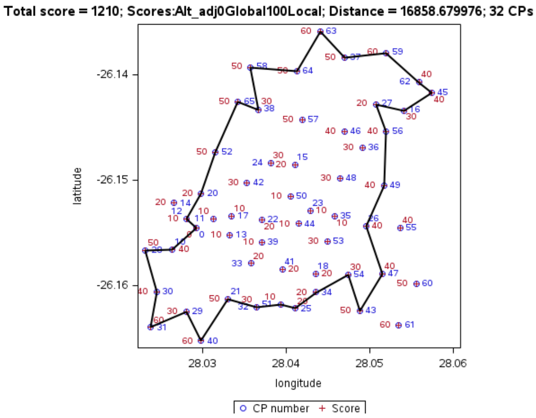
(j) 14 km



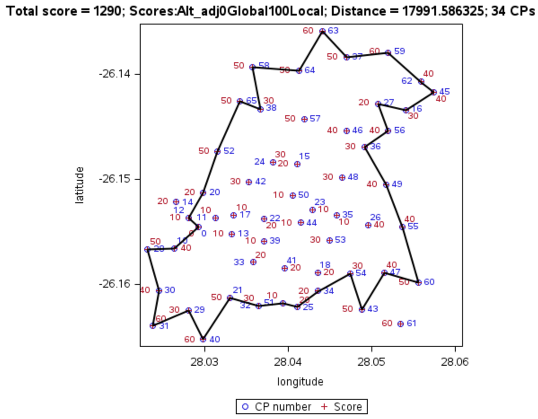
(k) 15 km



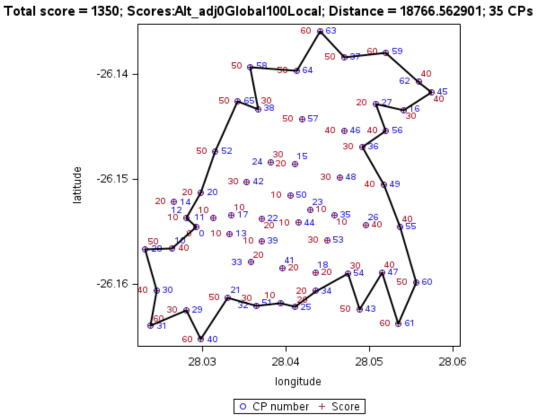
(l) 16 km



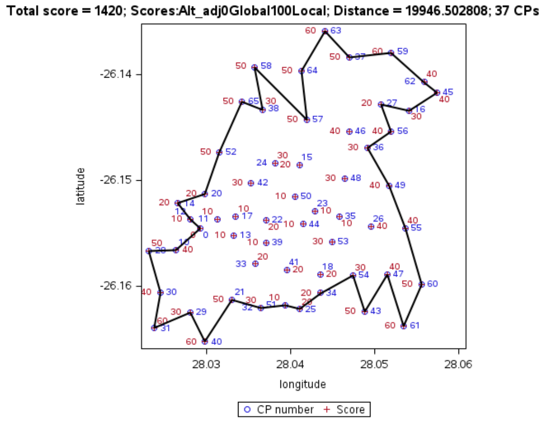
(m) 17 km



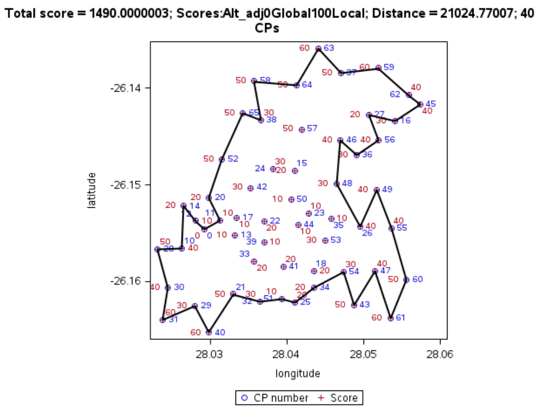
(n) 18 km



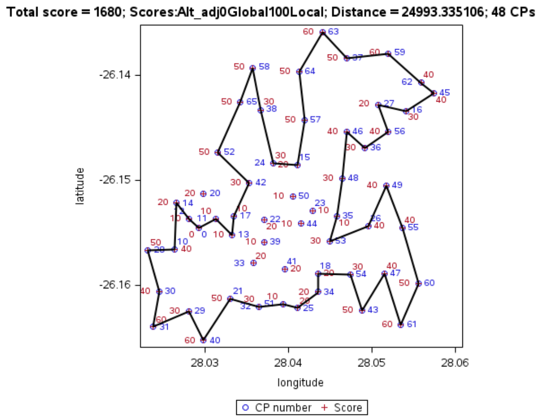
(o) 19 km



(p) 20 km

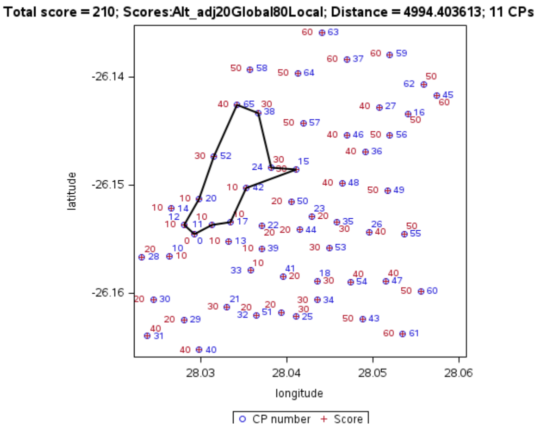


(q) 21,1 km

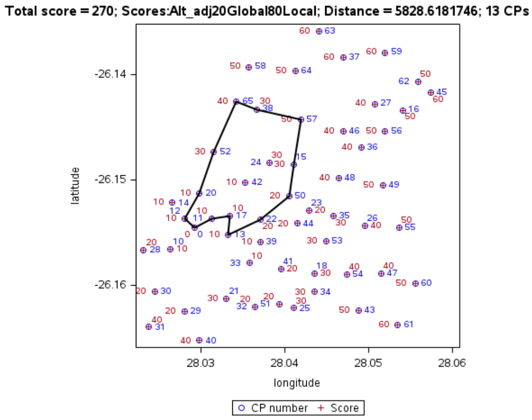


(r) 25 km

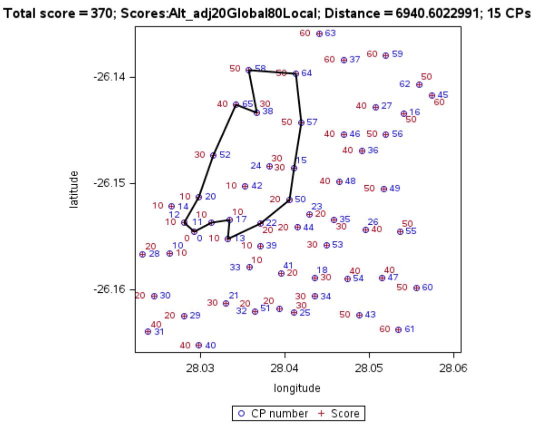
K.19 Alt_adj20G80L OP output



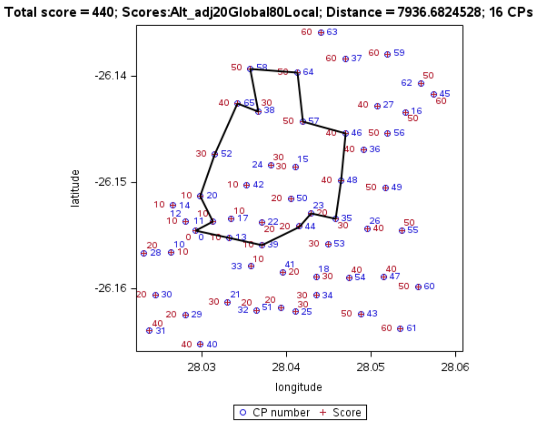
(a) 5 km



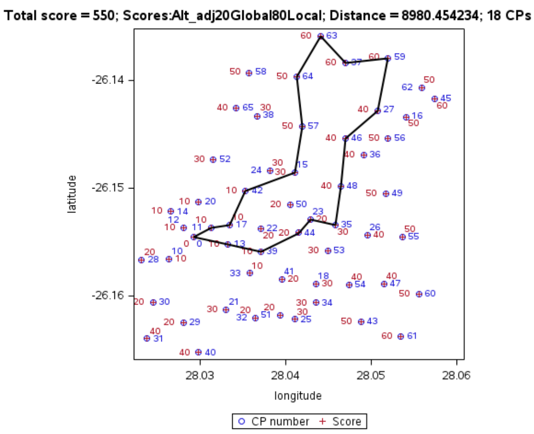
(b) 6 km



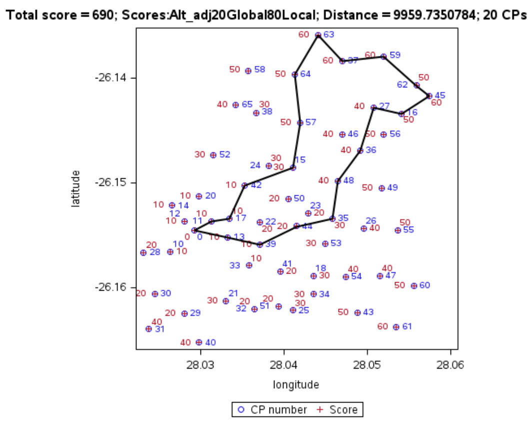
(c) 7 km



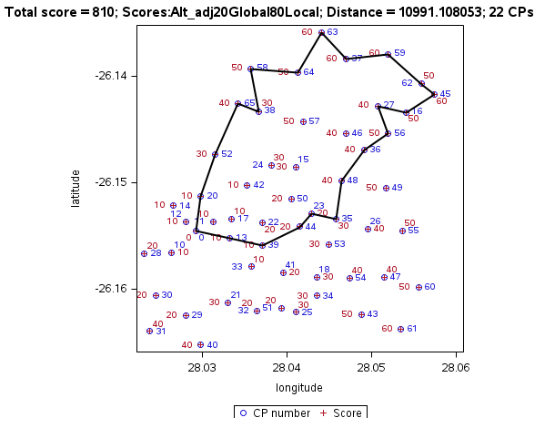
(d) 8 km



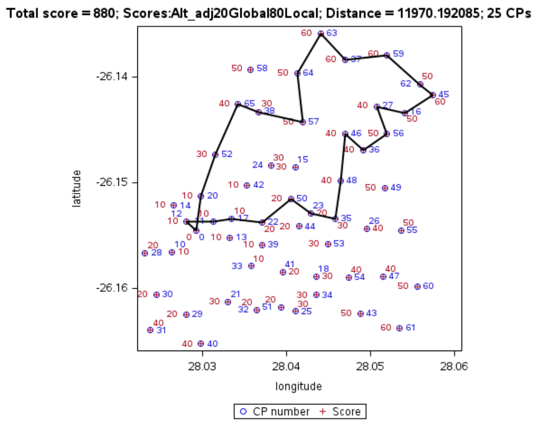
(e) 9 km



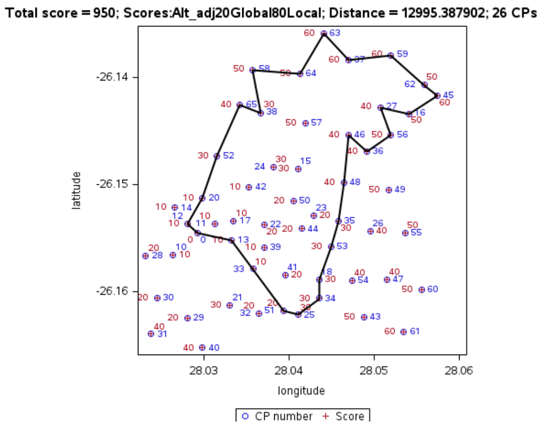
(f) 10 km



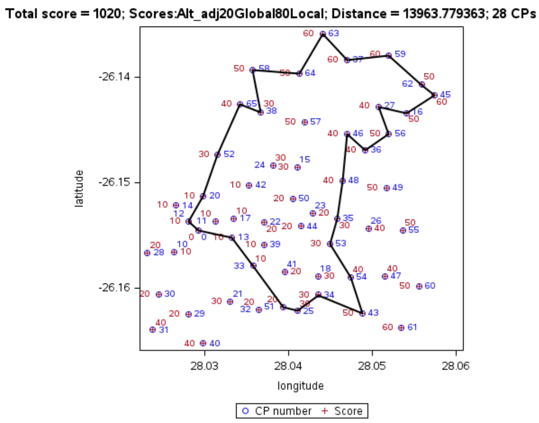
(g) 11 km



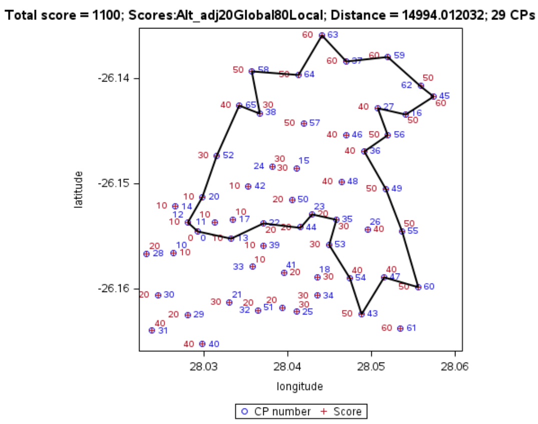
(h) 12 km



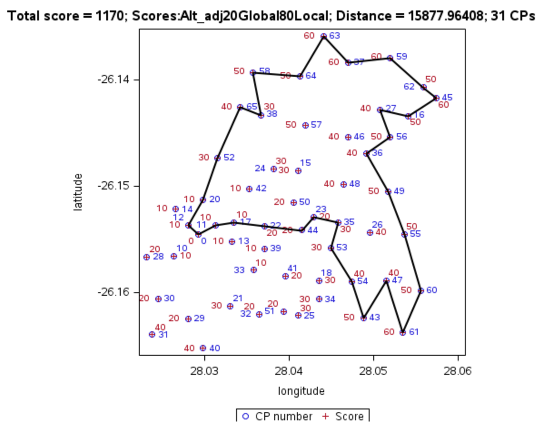
(i) 13 km



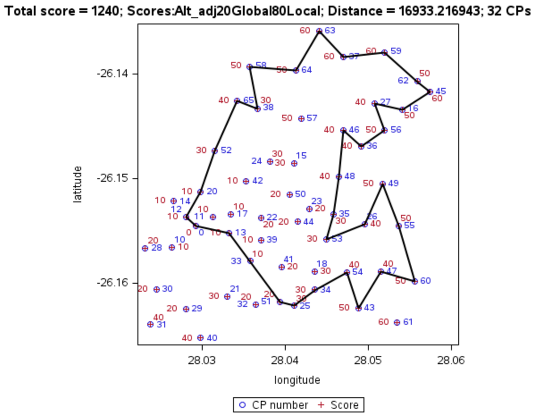
(j) 14 km



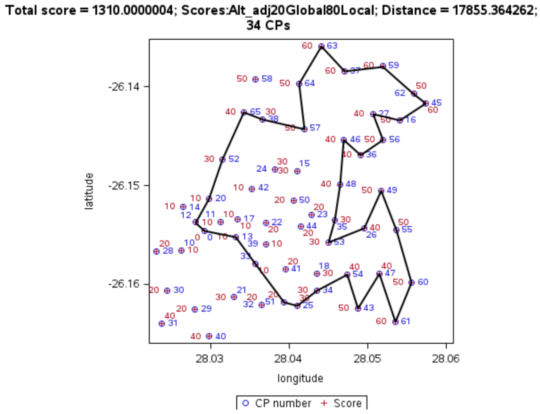
(k) 15 km



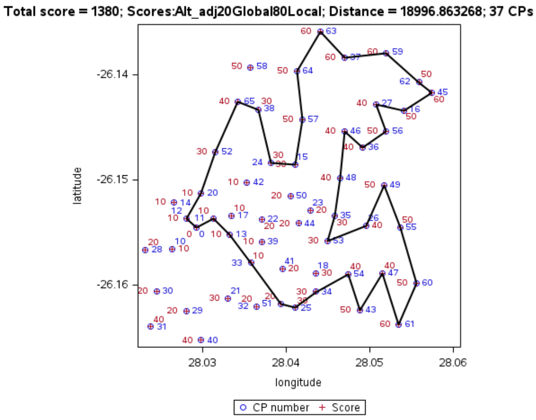
(l) 16 km



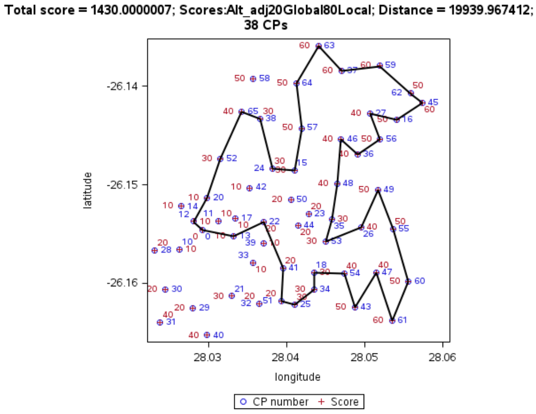
(m) 17 km



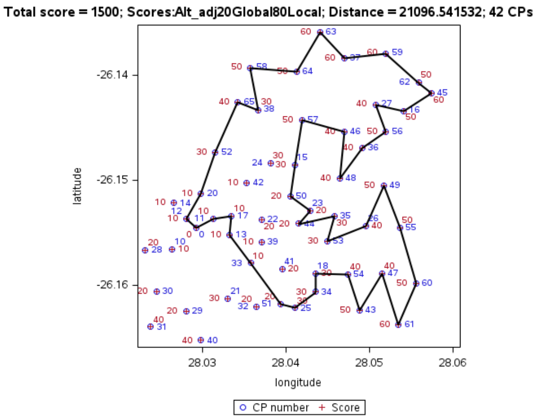
(n) 18 km



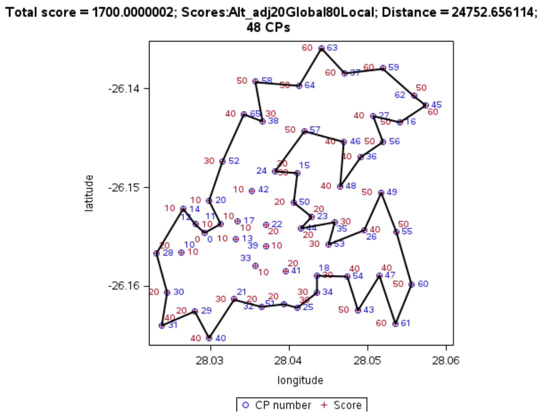
(o) 19 km



(p) 20 km

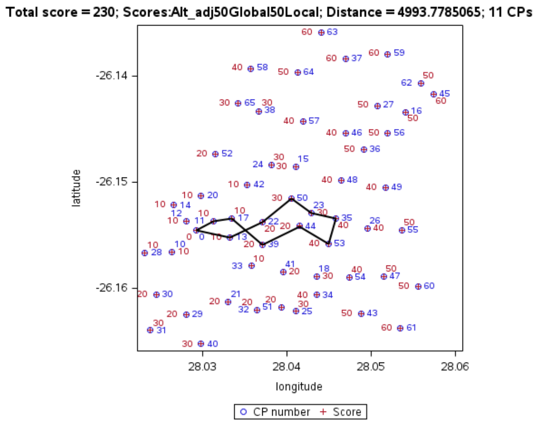


(q) 21,1 km

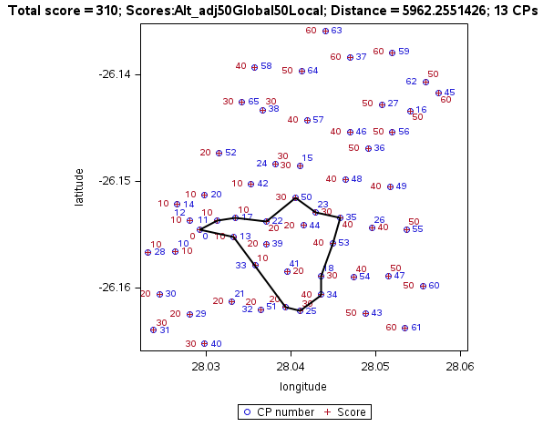


(r) 25 km

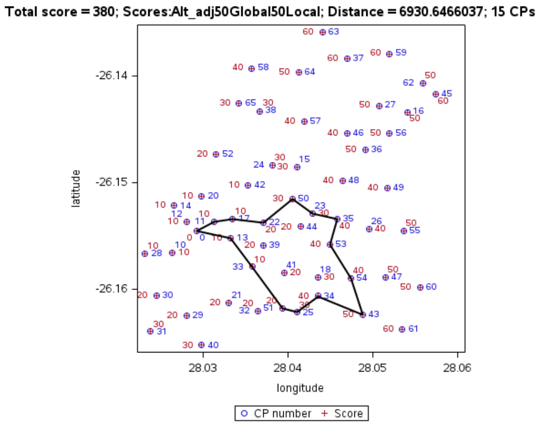
K.20 Alt_adj50G50L OP output



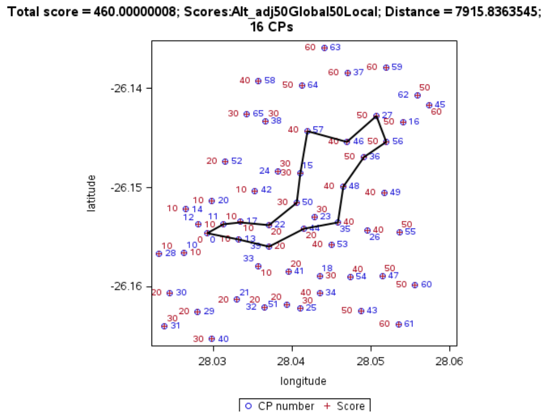
(a) 5 km



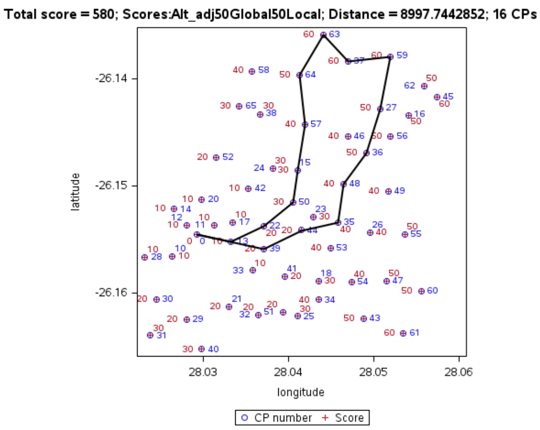
(b) 6 km



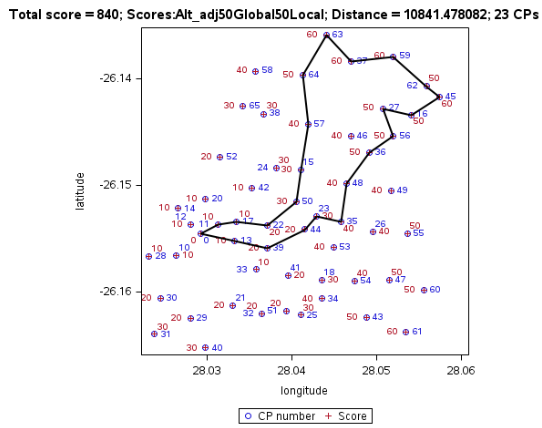
(c) 7 km



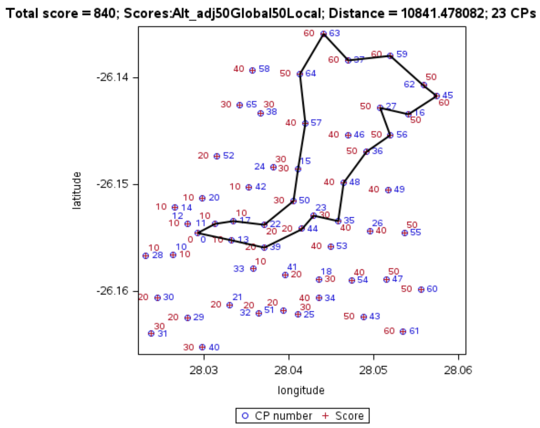
(d) 8 km



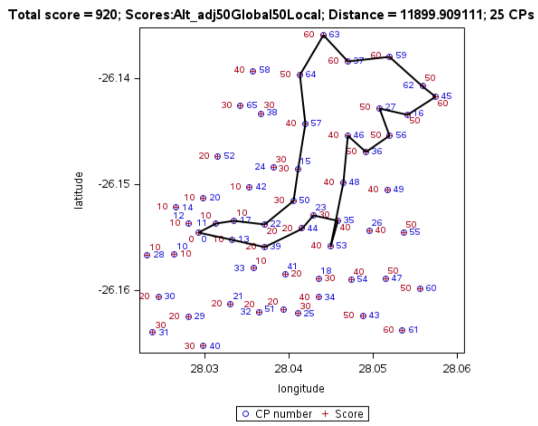
(e) 9 km



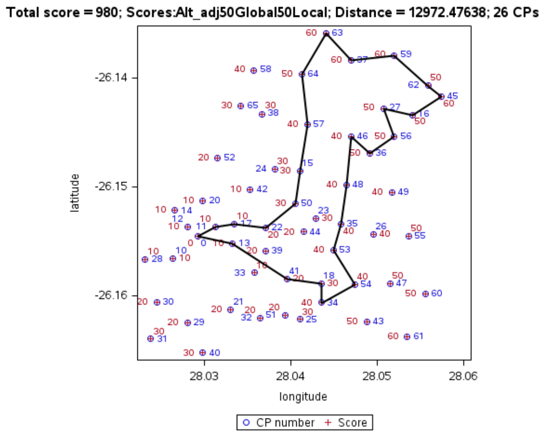
(f) 10 km



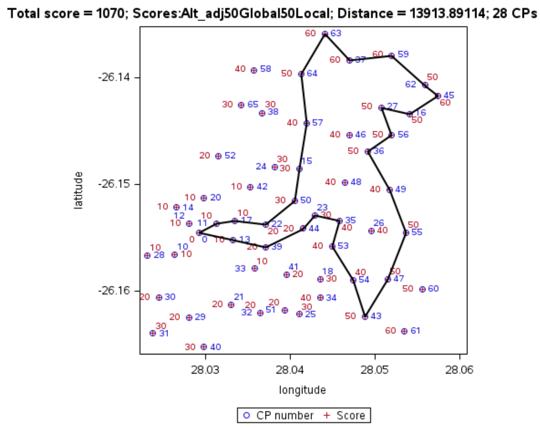
(g) 11 km



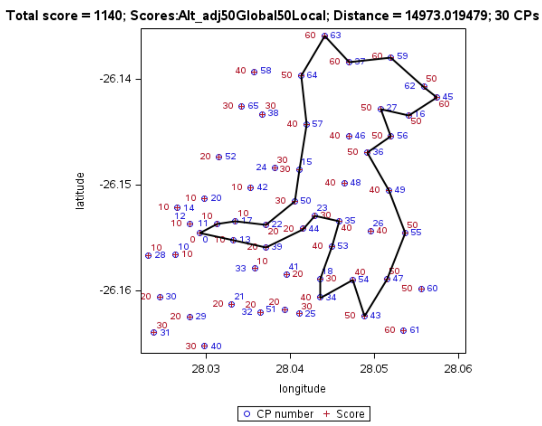
(h) 12 km



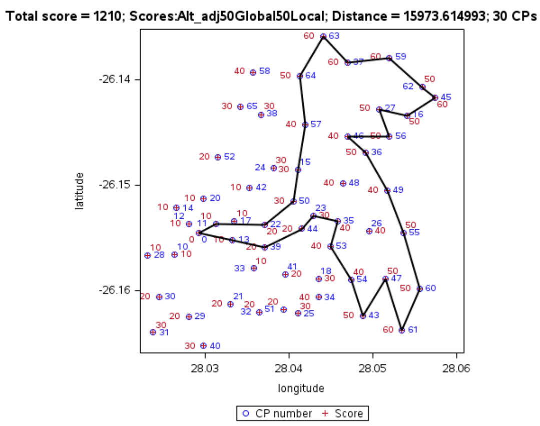
(i) 13 km



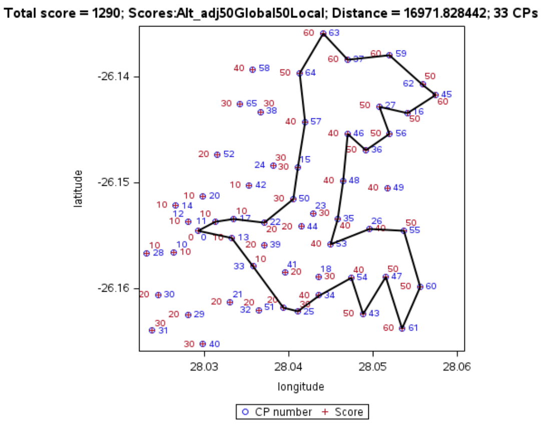
(j) 14 km



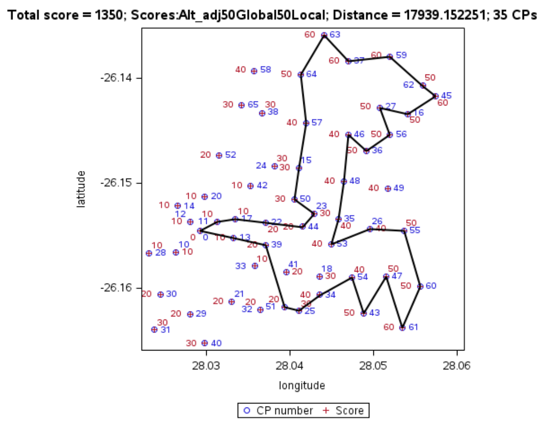
(k) 15 km



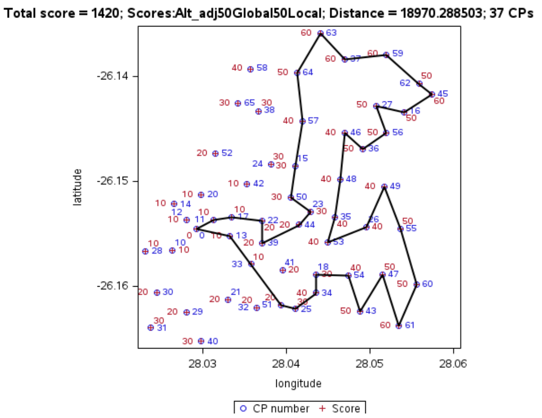
(l) 16 km



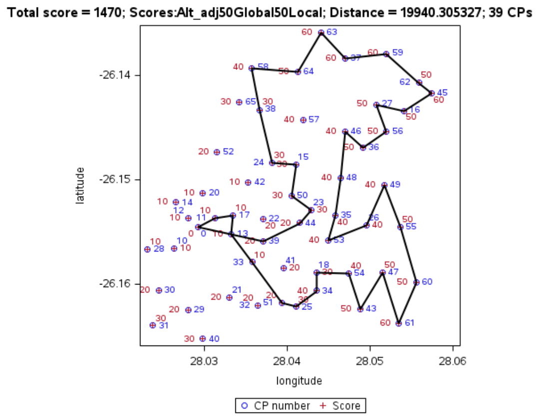
(m) 17 km



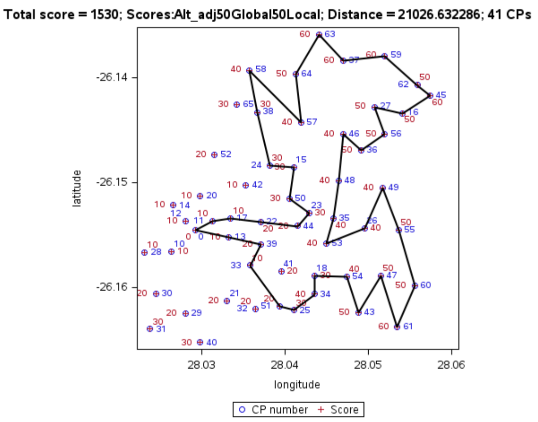
(n) 18 km



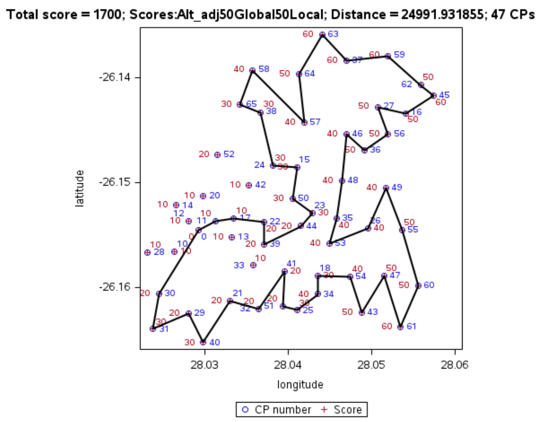
(o) 19 km



(p) 20 km

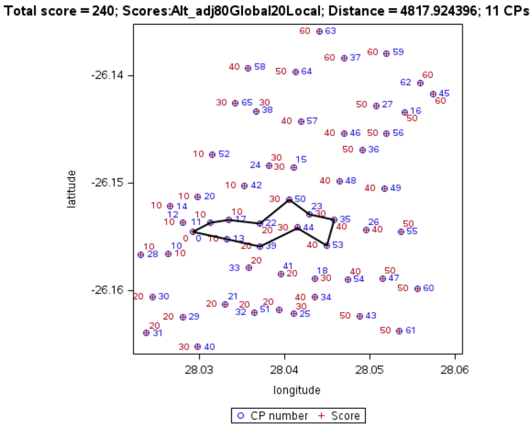


(q) 21,1 km

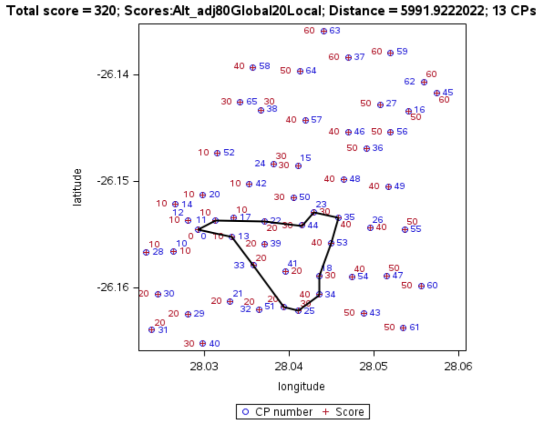


(r) 25 km

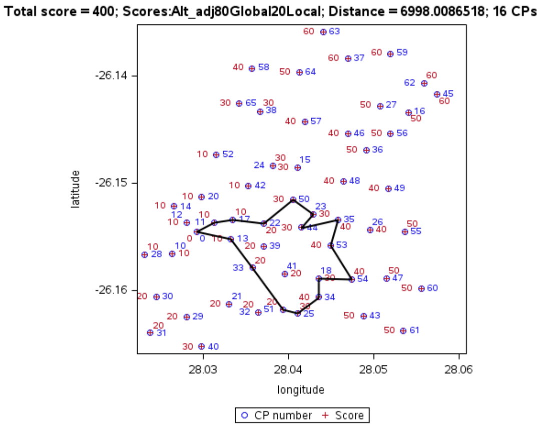
K.21 Alt_adj80G20L OP output



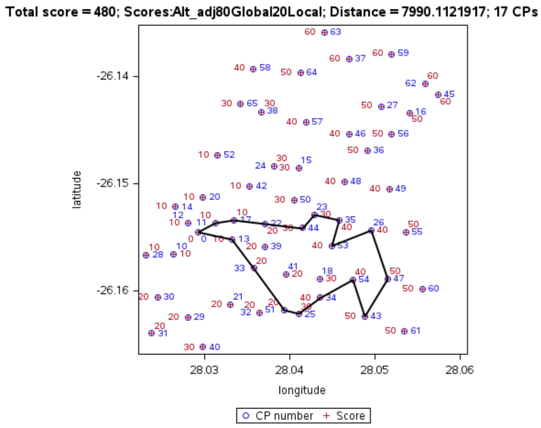
(a) 5 km



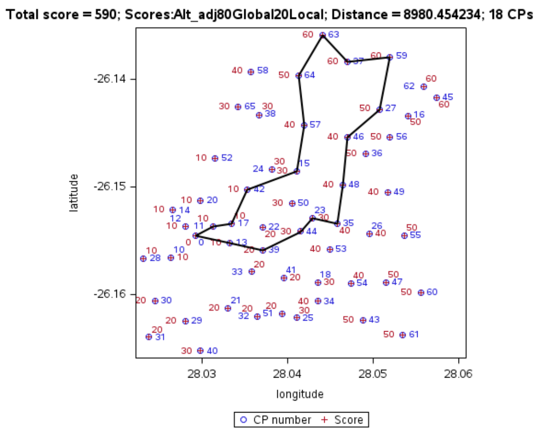
(b) 6 km



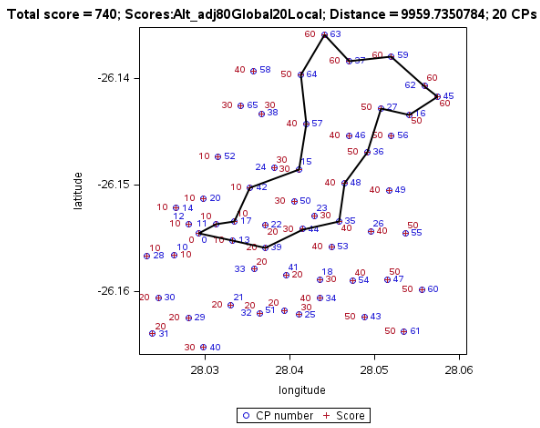
(c) 7 km



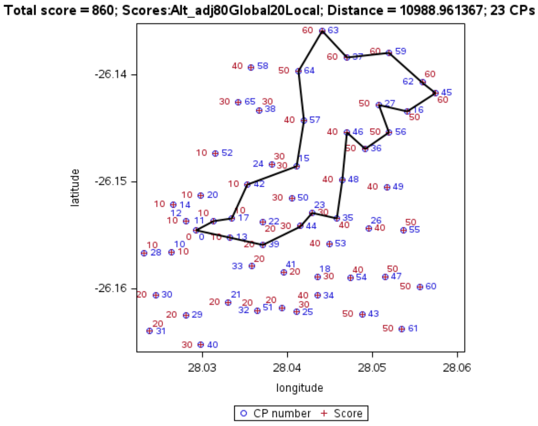
(d) 8 km



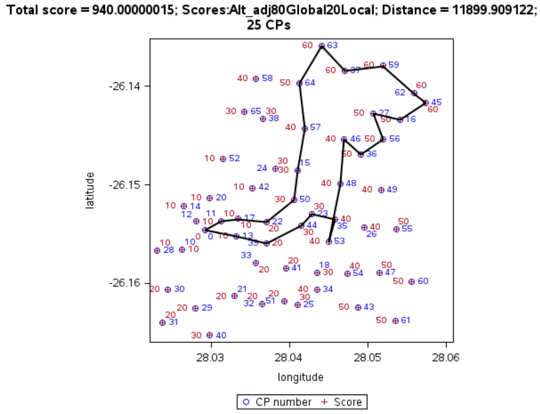
(e) 9 km



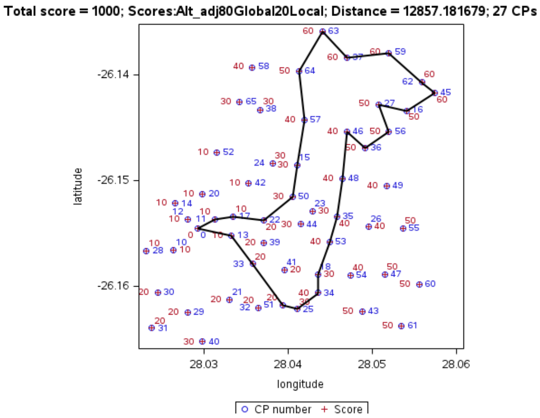
(f) 10 km



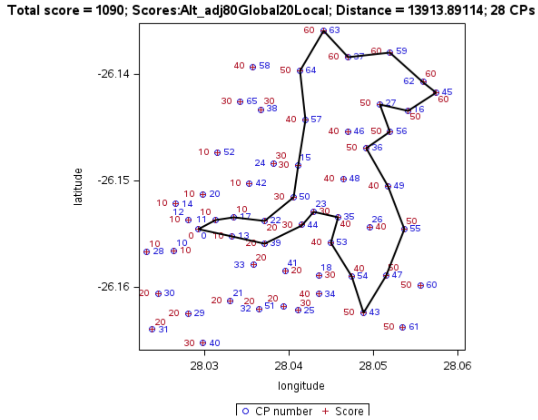
(g) 11 km



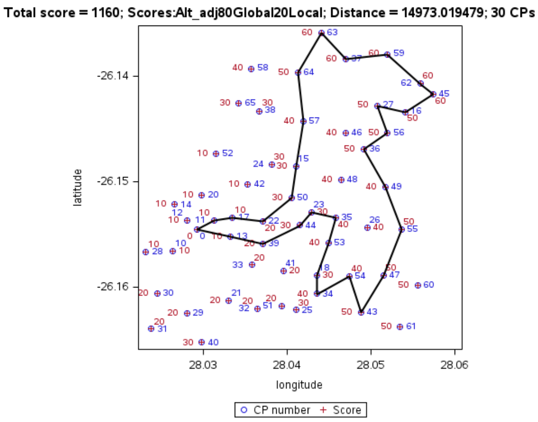
(h) 12 km



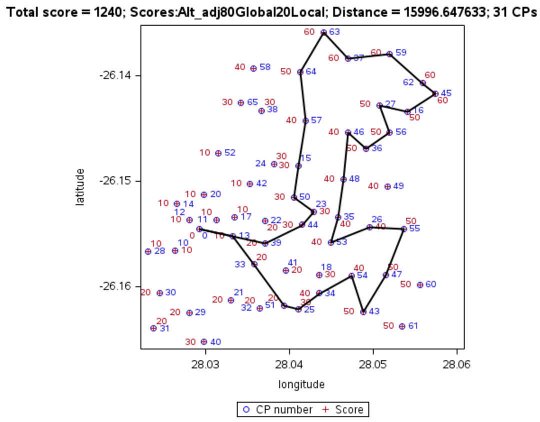
(i) 13 km



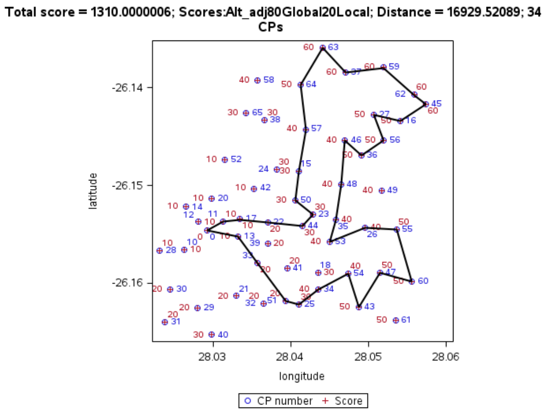
(j) 14 km



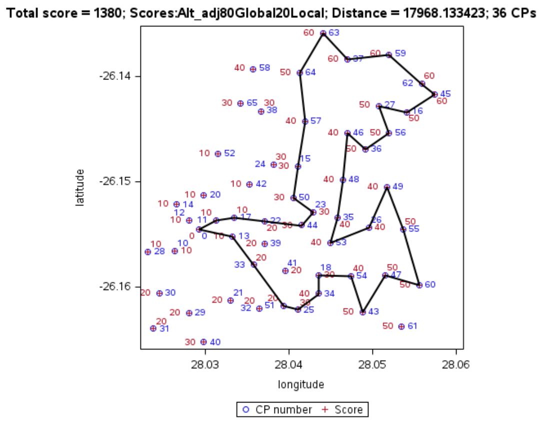
(k) 15 km



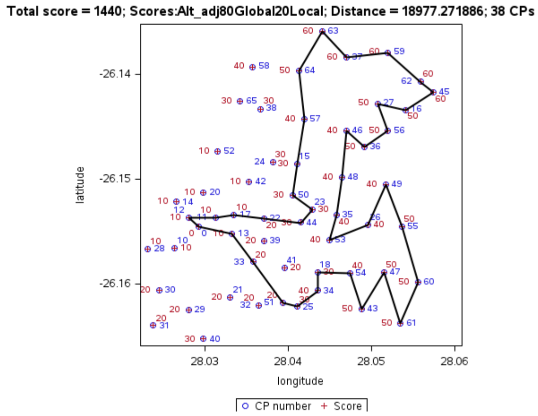
(l) 16 km



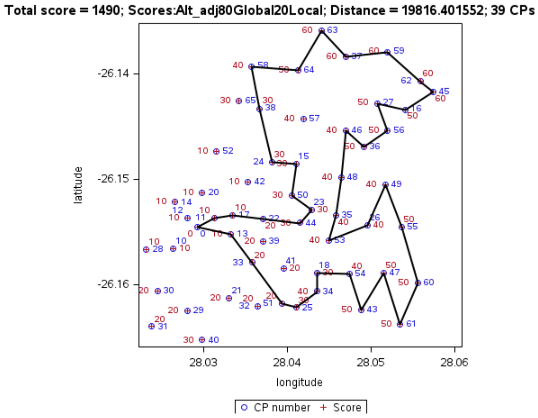
(m) 17 km



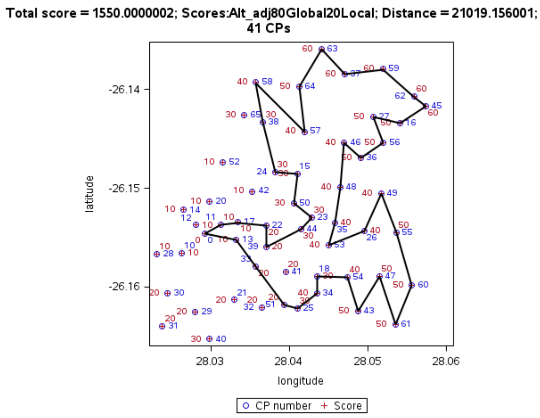
(n) 18 km



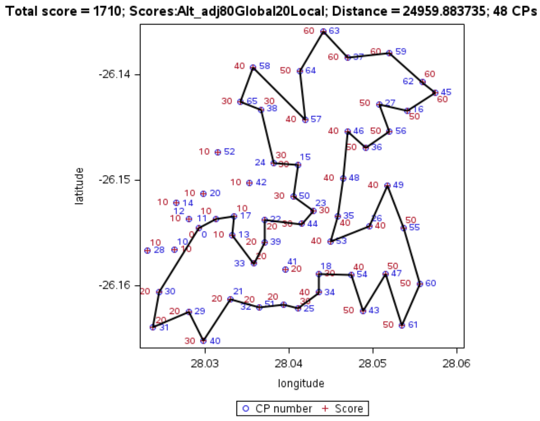
(o) 19 km



(p) 20 km

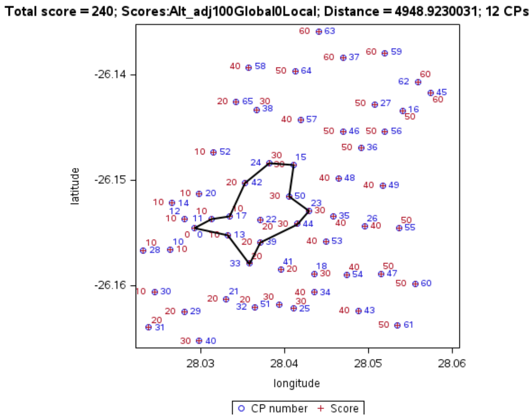


(q) 21,1 km

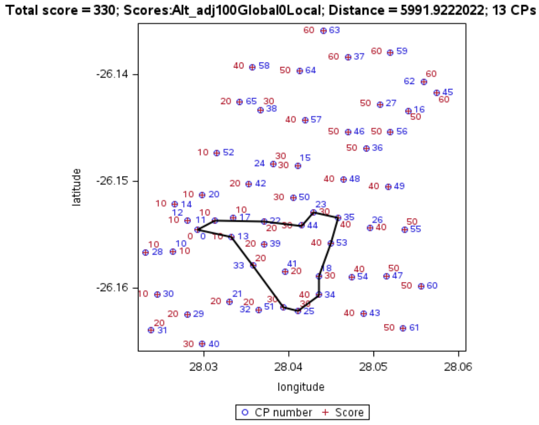


(r) 25 km

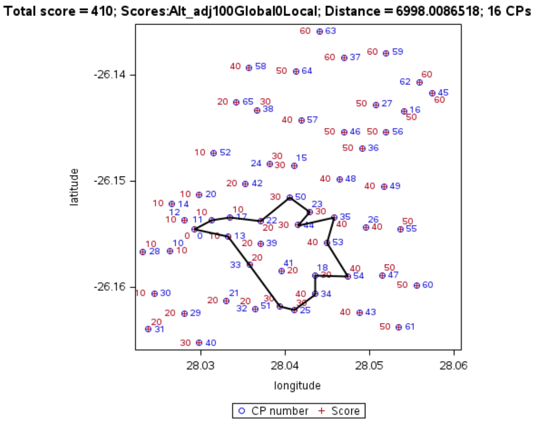
K.22 Alt_adj100G0L OP output



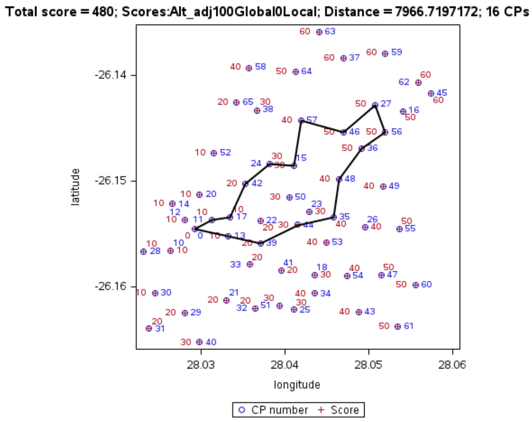
(a) 5 km



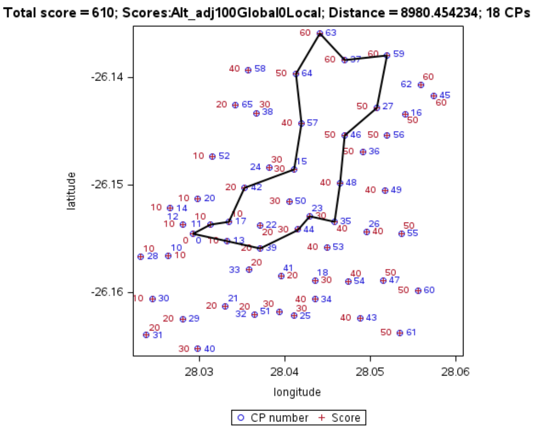
(b) 6 km



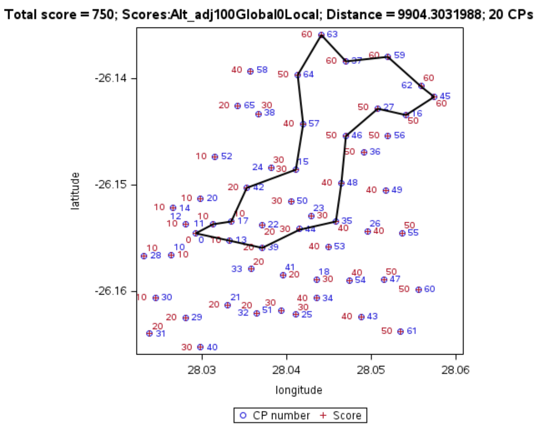
(c) 7 km



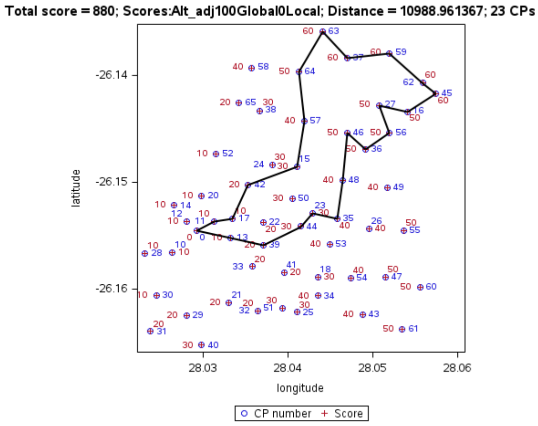
(d) 8 km



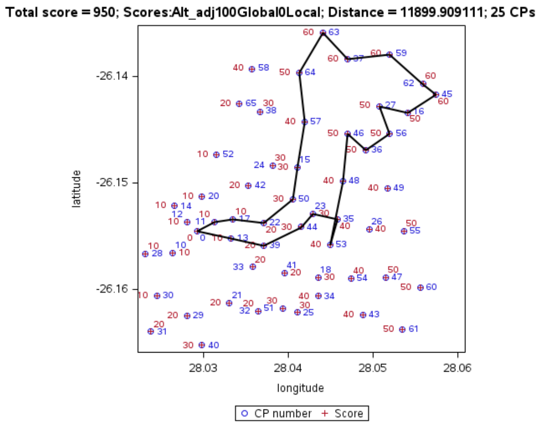
(e) 9 km



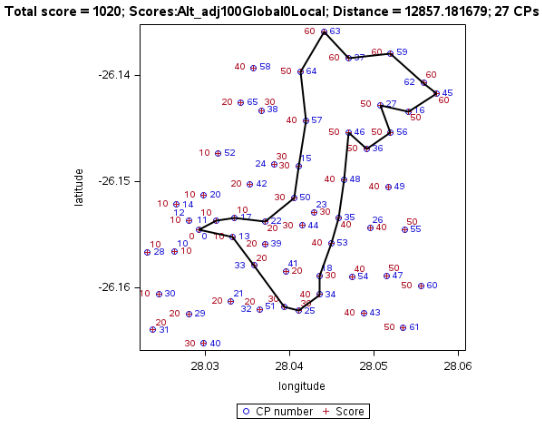
(f) 10 km



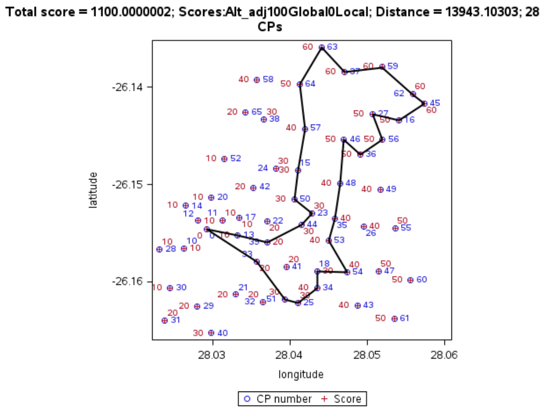
(g) 11 km



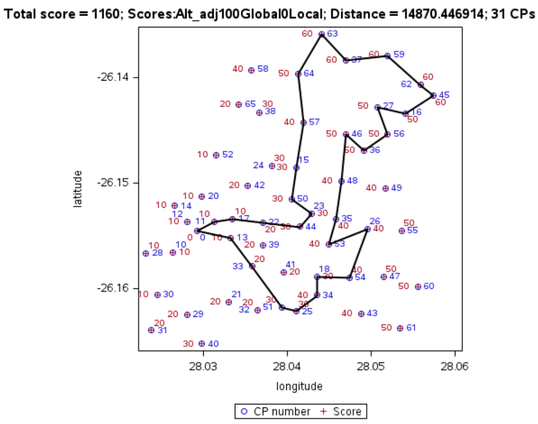
(h) 12 km



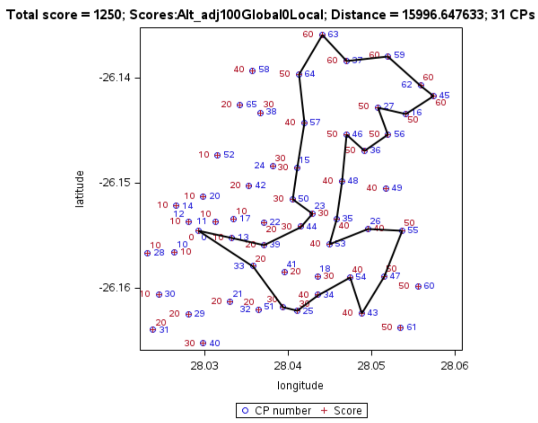
(i) 13 km



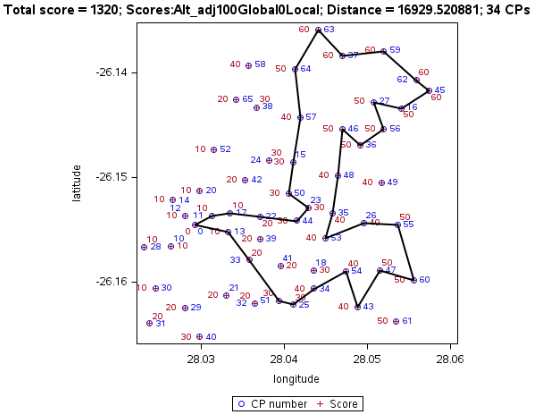
(j) 14 km



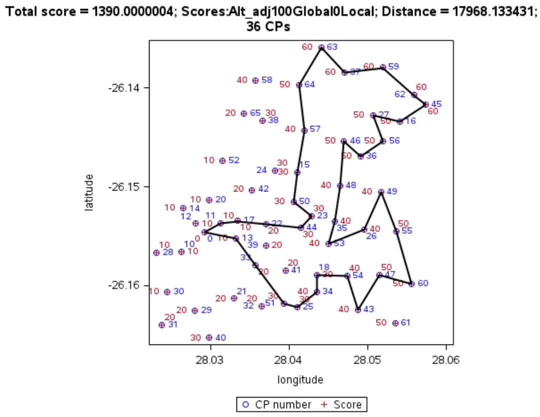
(k) 15 km



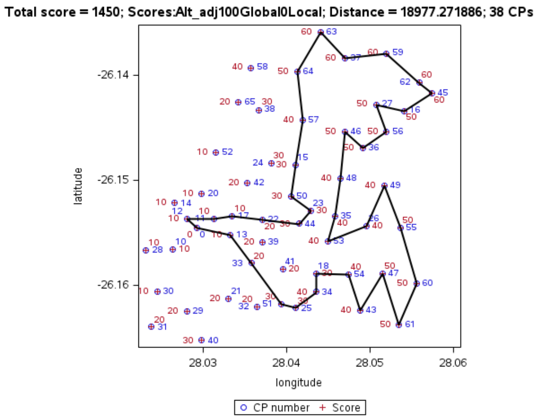
(l) 16 km



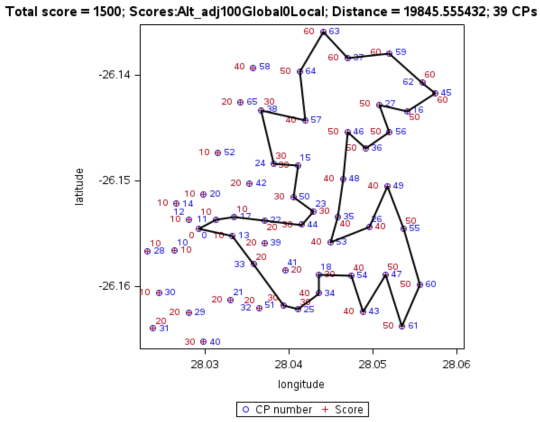
(m) 17 km



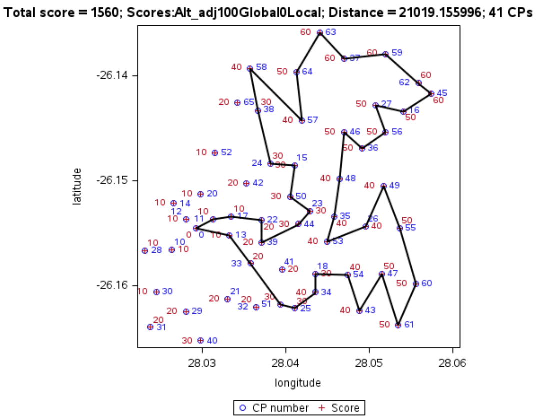
(n) 18 km



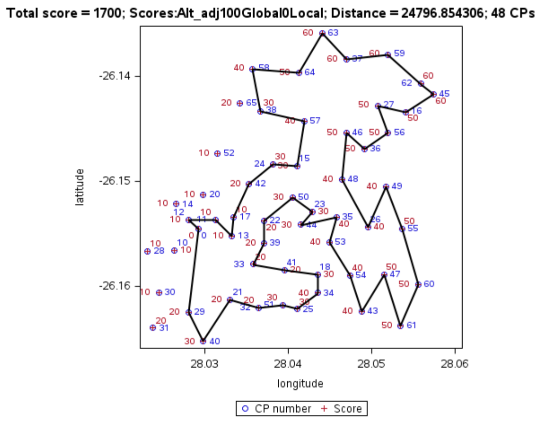
(o) 19 km



(p) 20 km

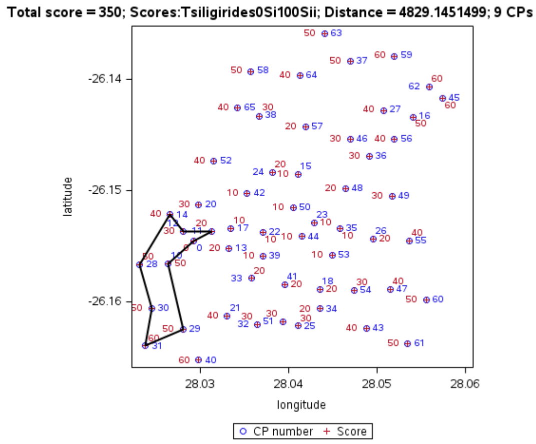


(q) 21,1 km

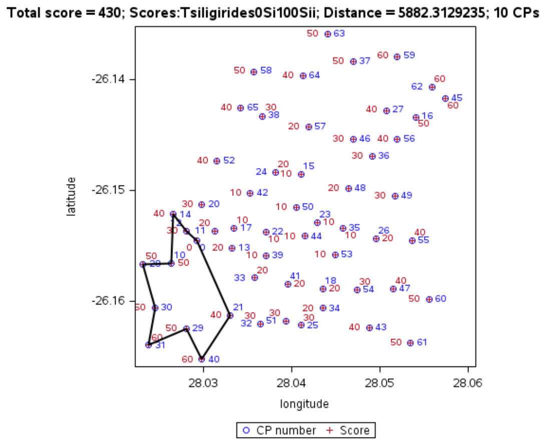


(r) 25 km

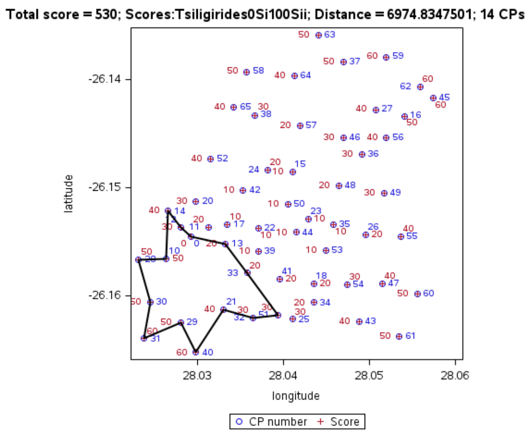
K.23 TS0si100sii OP output



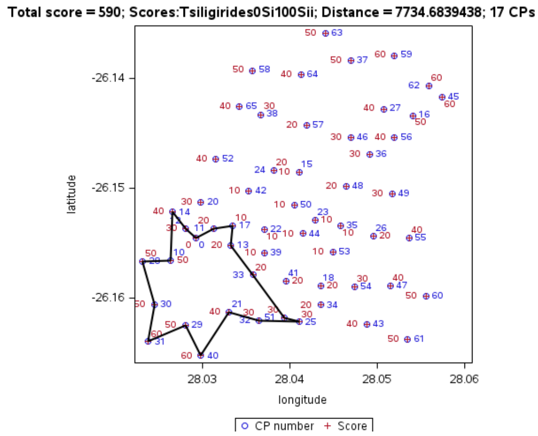
(a) 5 km



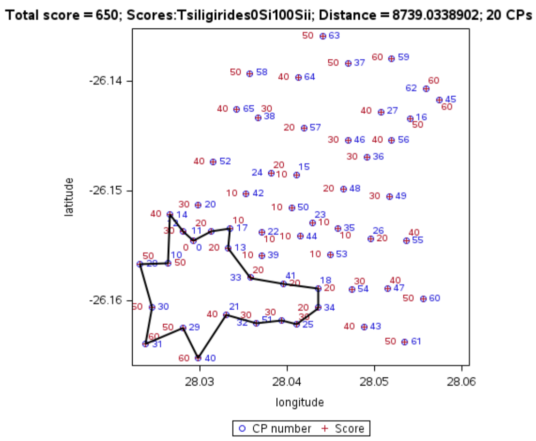
(b) 6 km



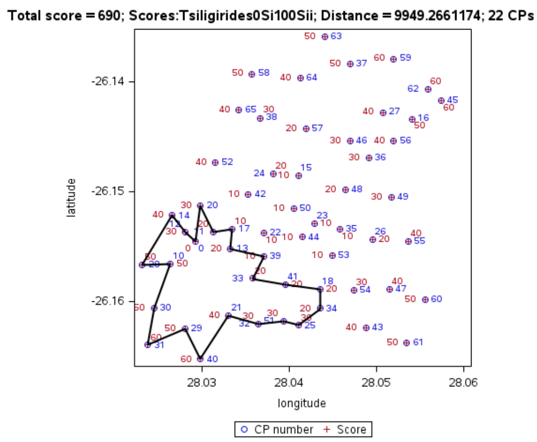
(c) 7 km



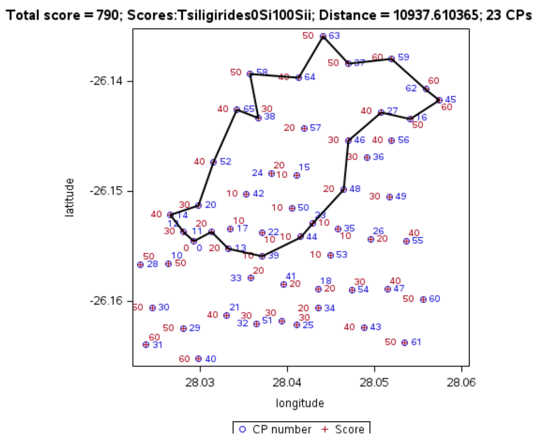
(d) 8 km



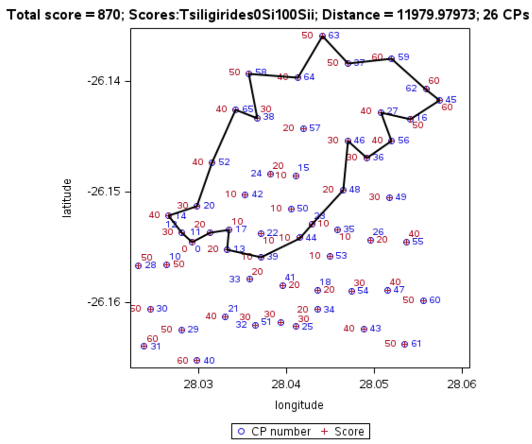
(e) 9 km



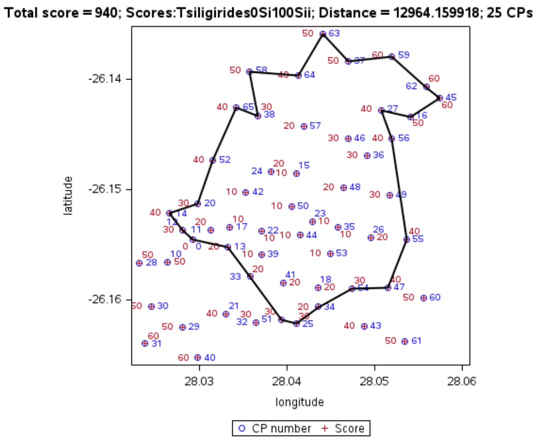
(f) 10 km



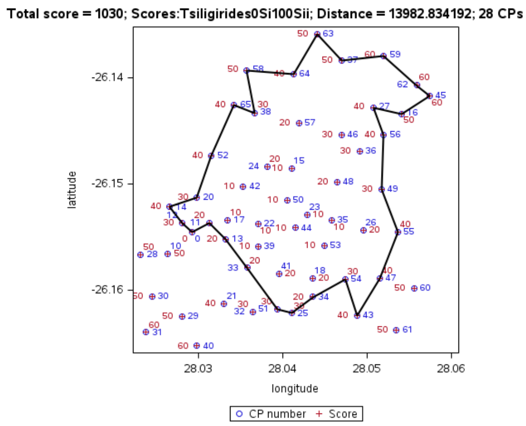
(g) 11 km



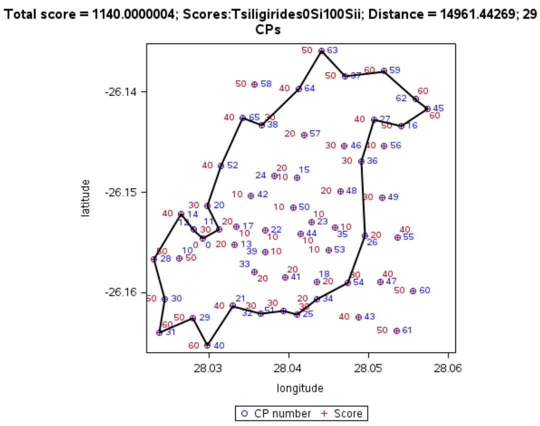
(h) 12 km



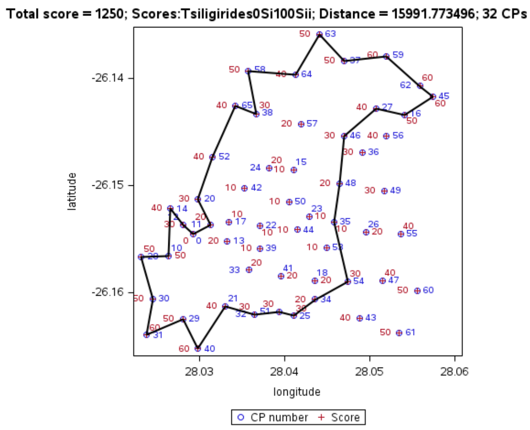
(i) 13 km



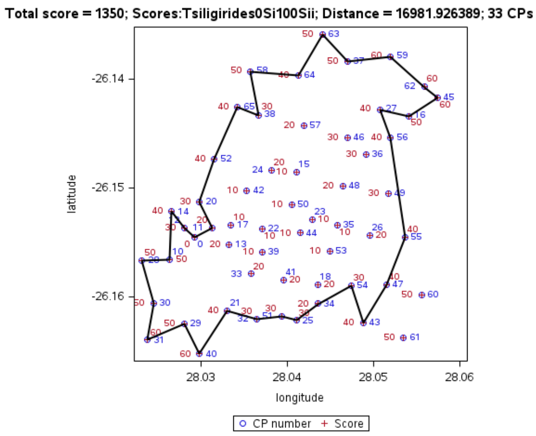
(j) 14 km



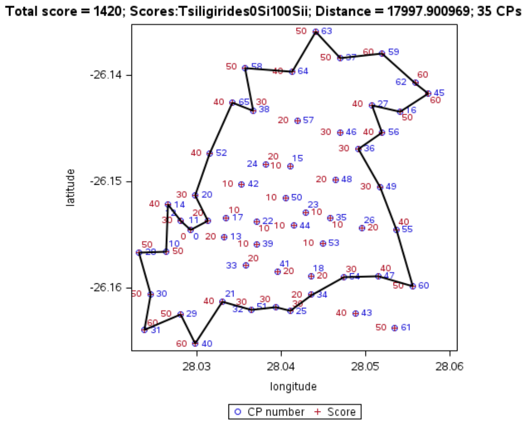
(k) 15 km



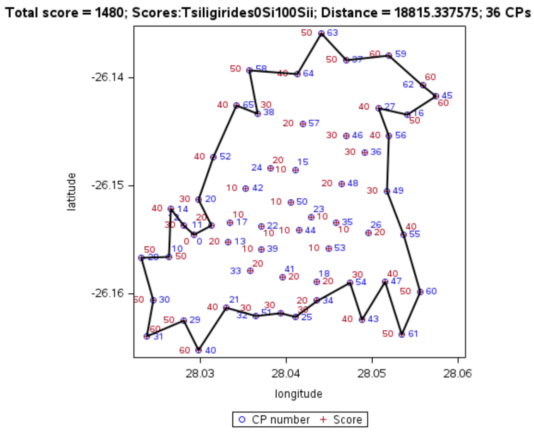
(l) 16 km



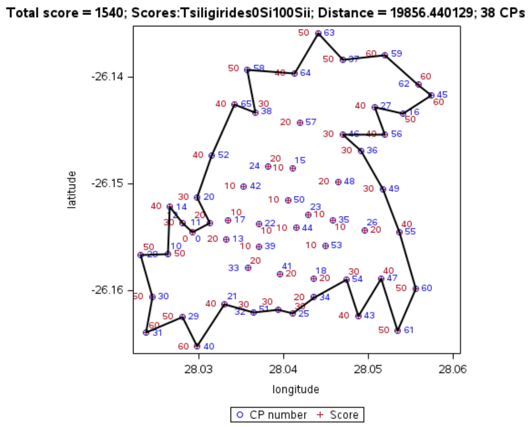
(m) 17 km



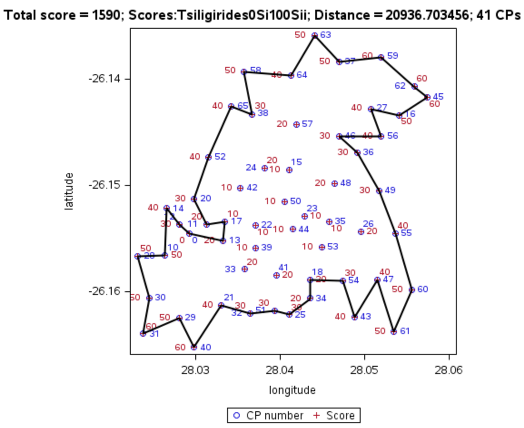
(n) 18 km



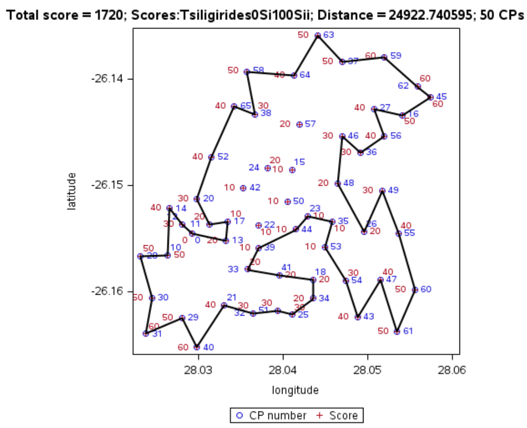
(o) 19 km



(p) 20 km

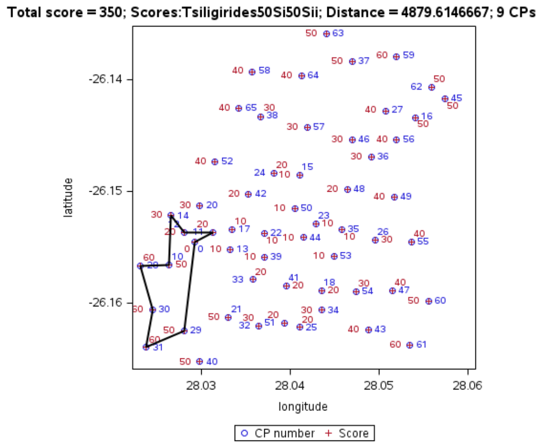


(q) 21,1 km

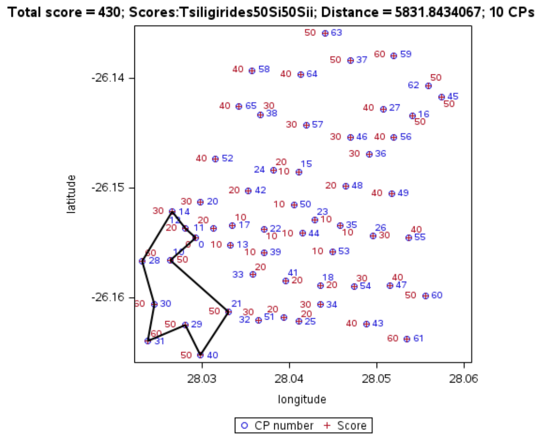


(r) 25 km

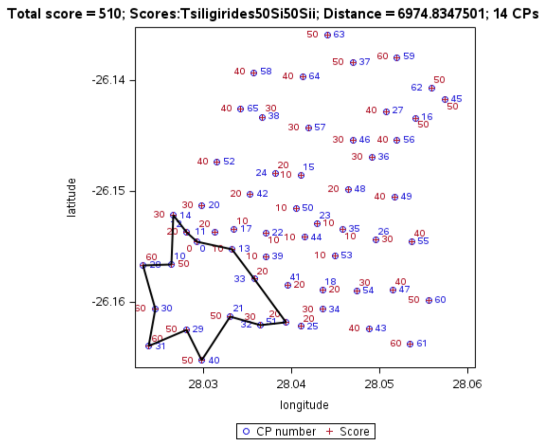
K.24 TS50si50sii OP output



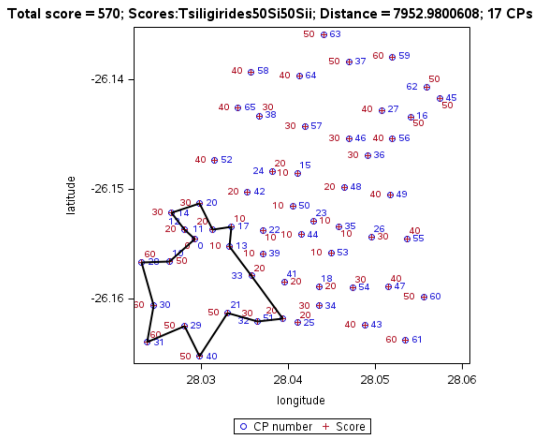
(a) 5 km



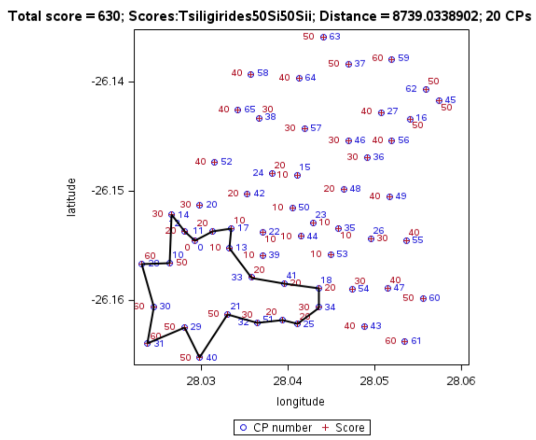
(b) 6 km



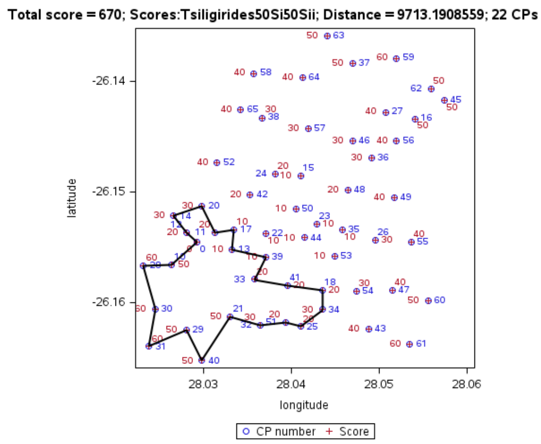
(c) 7 km



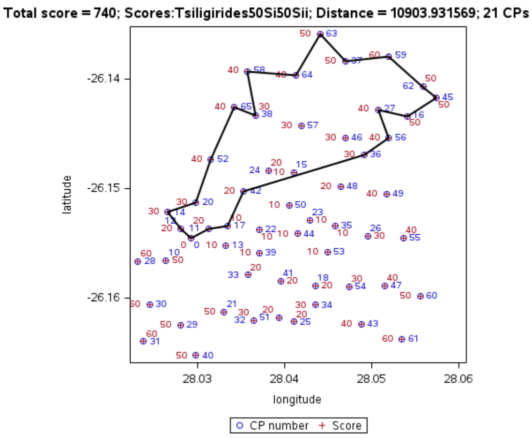
(d) 8 km



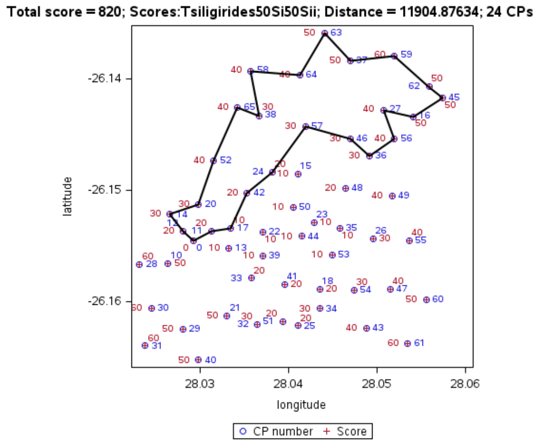
(e) 9 km



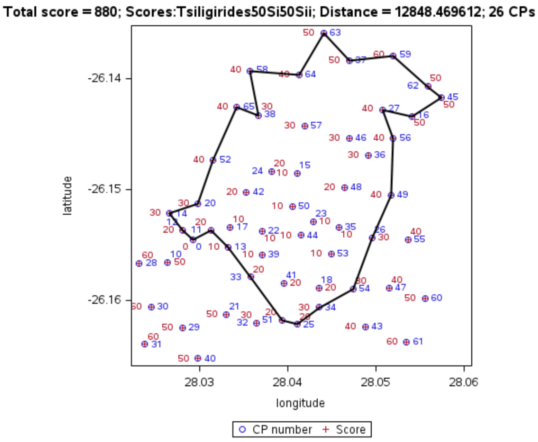
(f) 10 km



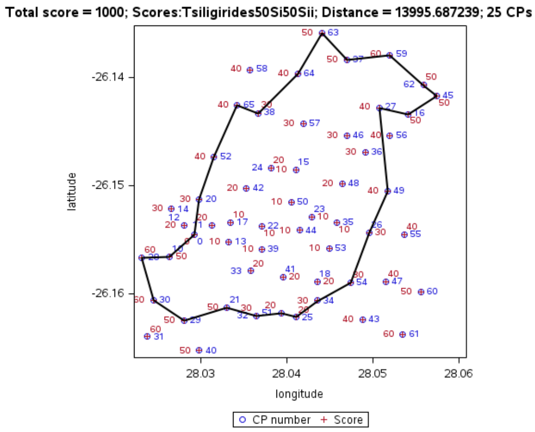
(g) 11 km



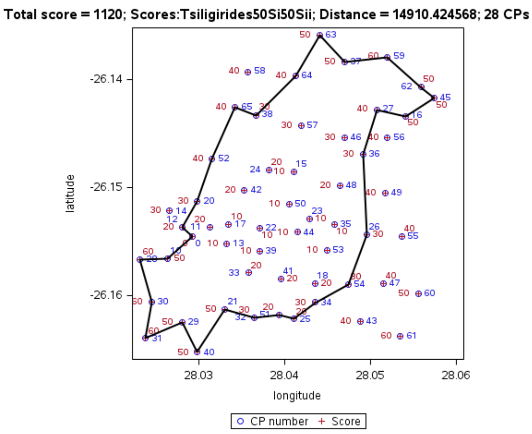
(h) 12 km



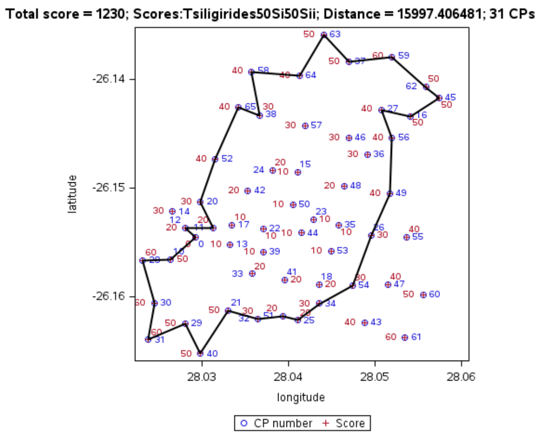
(i) 13 km



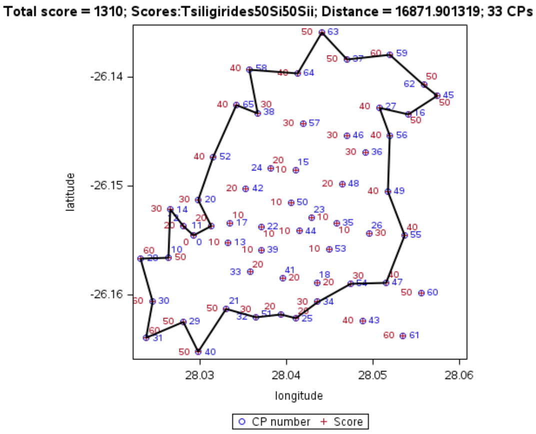
(j) 14 km



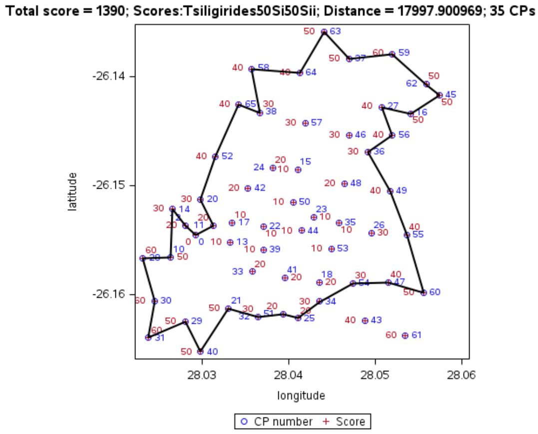
(k) 15 km



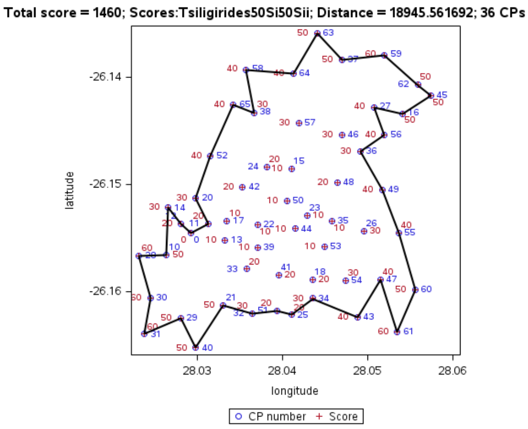
(l) 16 km



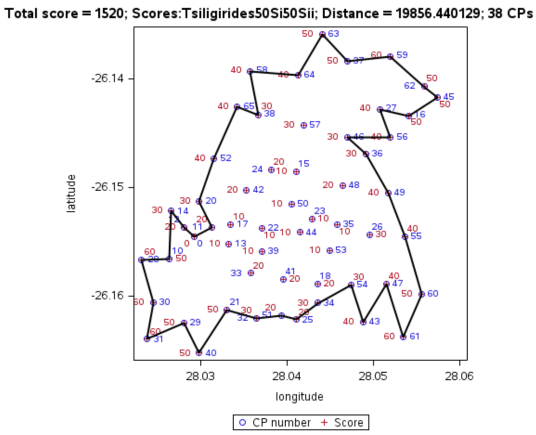
(m) 17 km



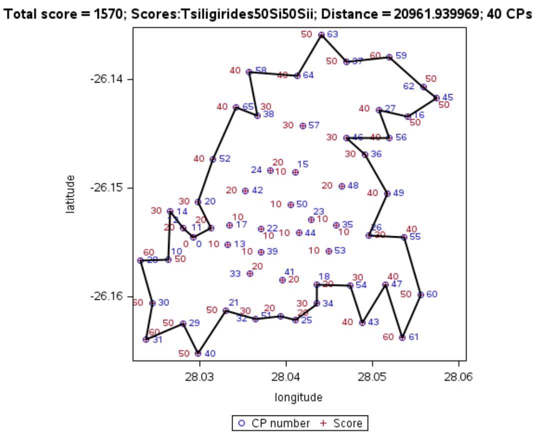
(n) 18 km



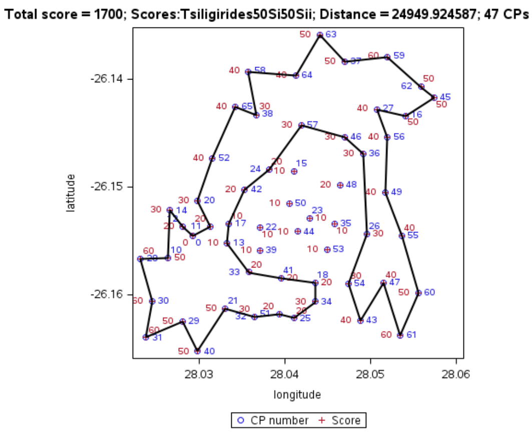
(o) 19 km



(p) 20 km

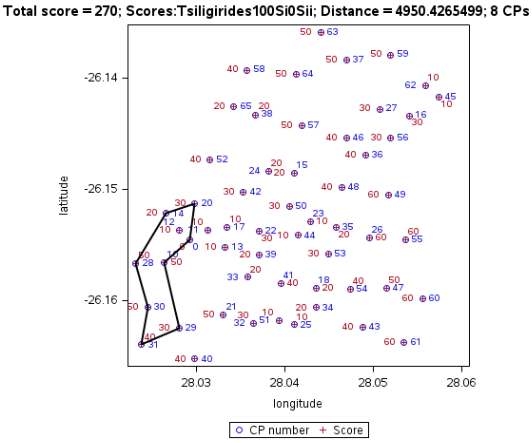


(q) 21,1 km

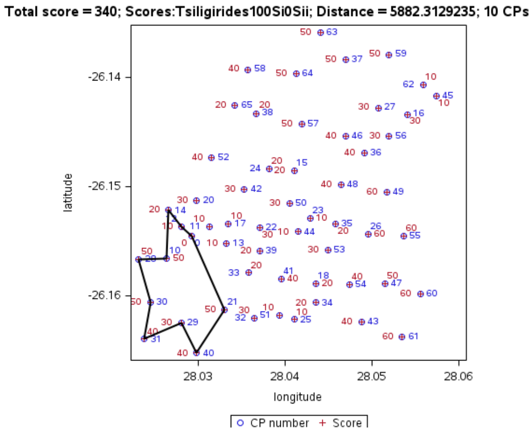


(r) 25 km

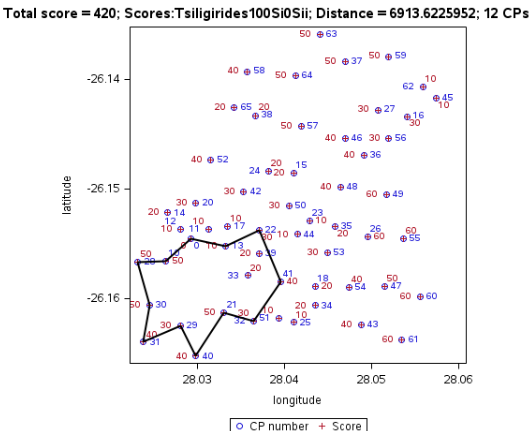
K.25 TS100si0sii OP output



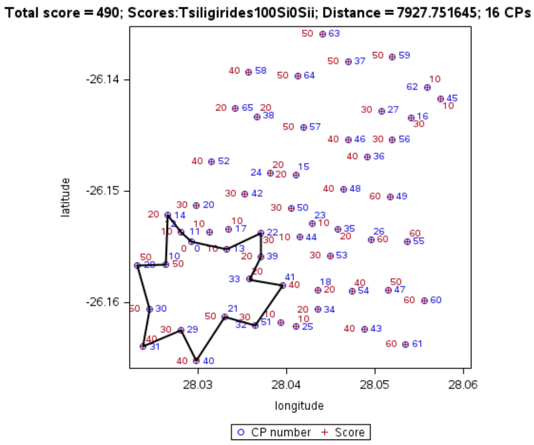
(a) 5 km



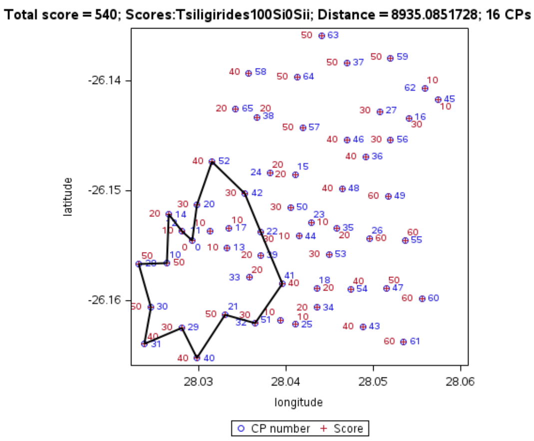
(b) 6 km



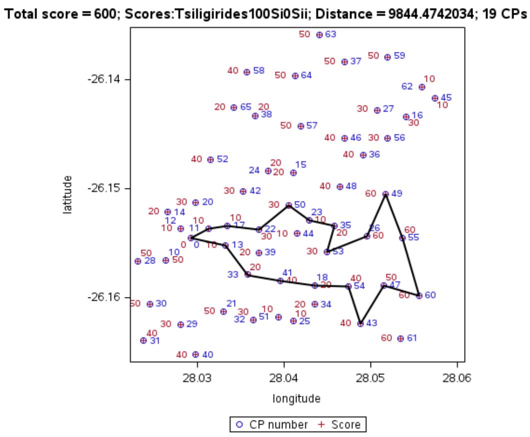
(c) 7 km



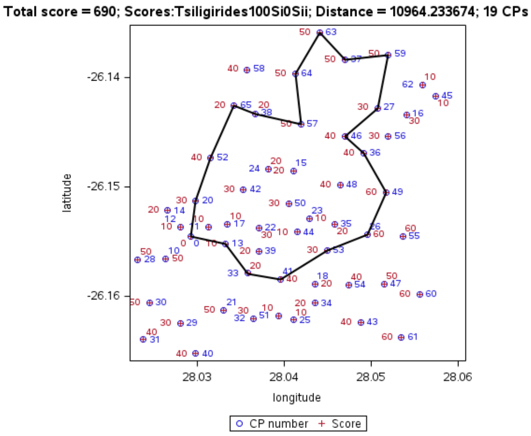
(d) 8 km



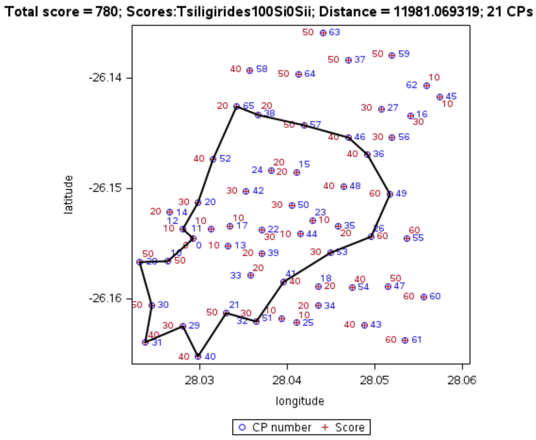
(e) 9 km



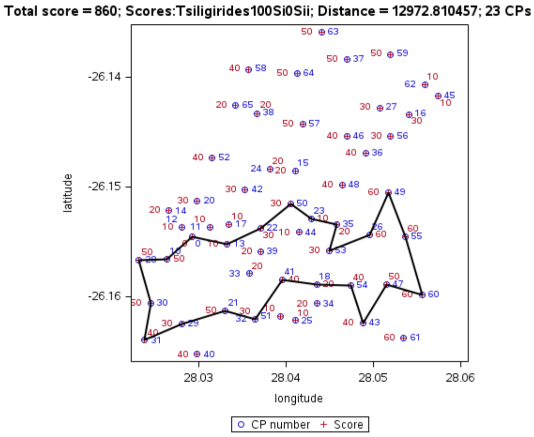
(f) 10 km



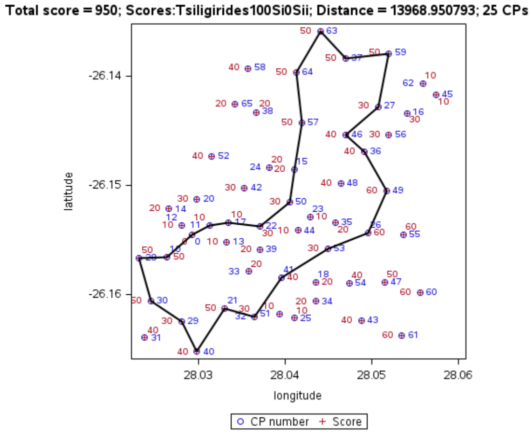
(g) 11 km



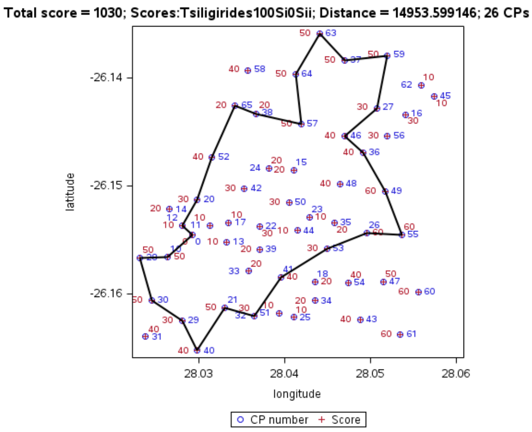
(h) 12 km



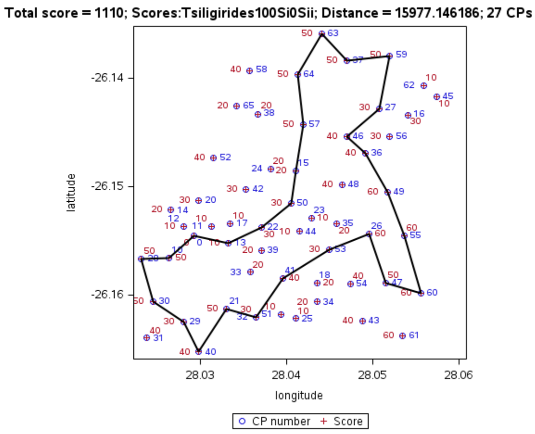
(i) 13 km



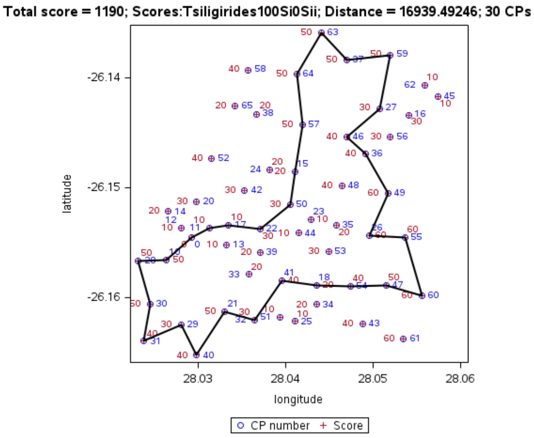
(j) 14 km



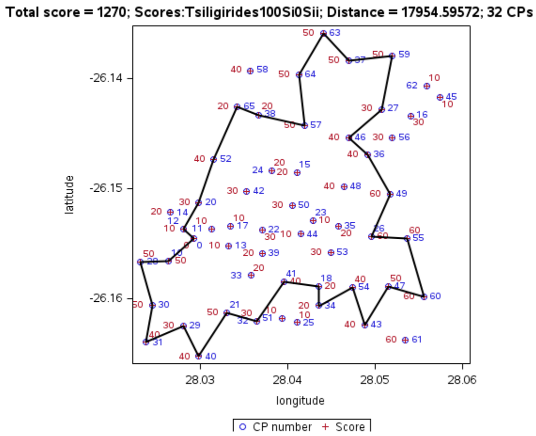
(k) 15 km



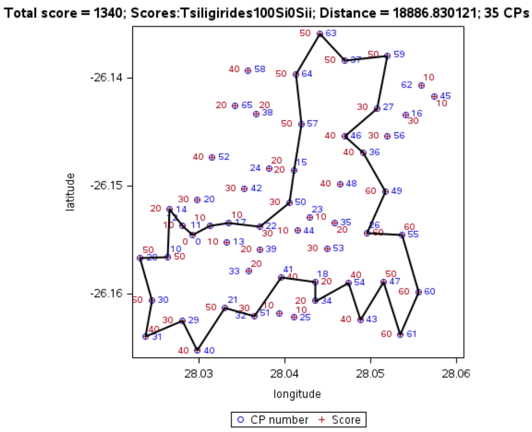
(l) 16 km



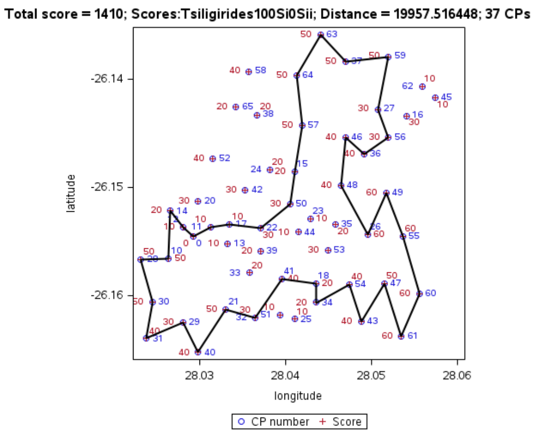
(m) 17 km



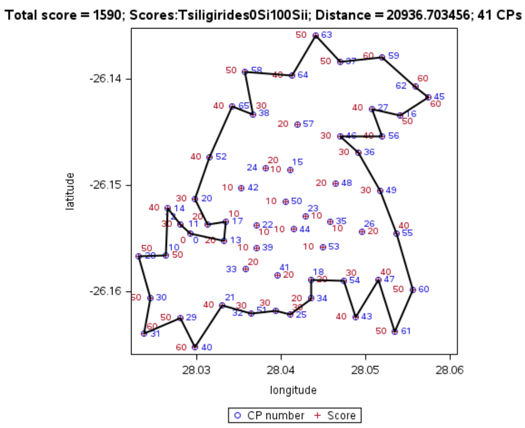
(n) 18 km



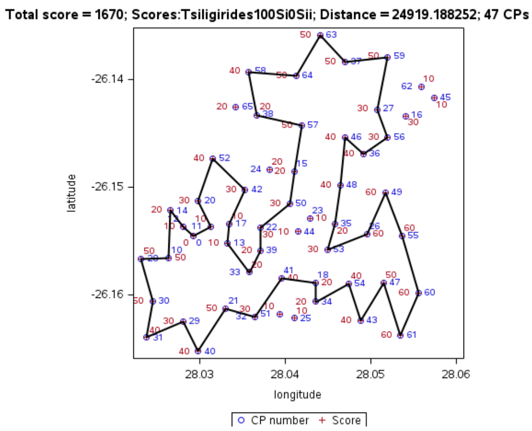
(o) 19 km



(p) 20 km

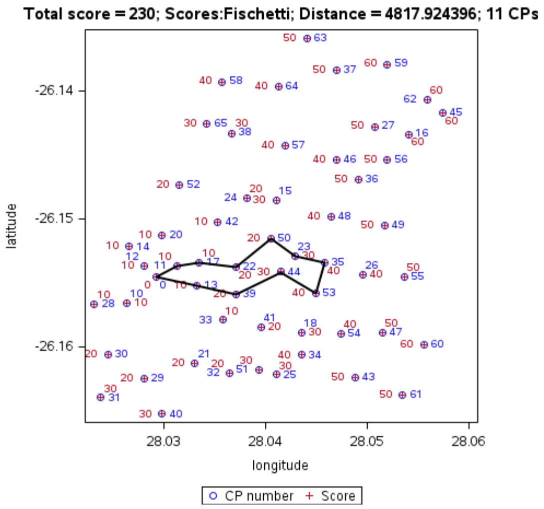


(q) 21,1 km

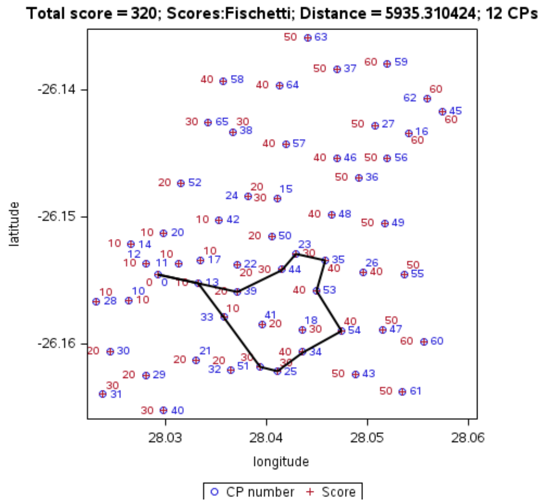


(r) 25 km

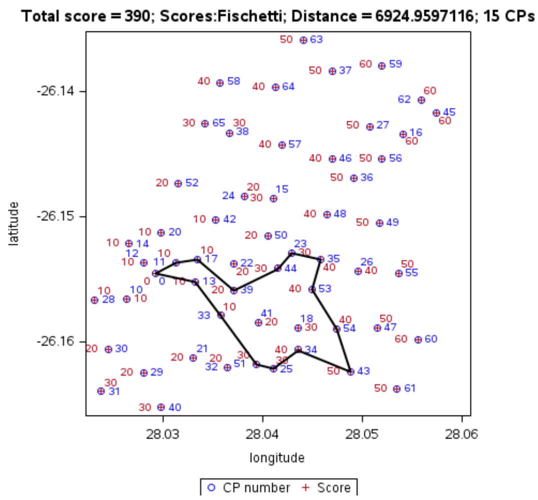
K.26 Fischetti OP output



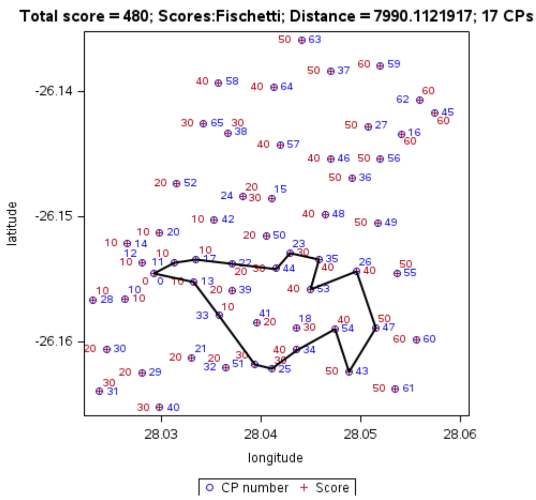
(a) 5 km



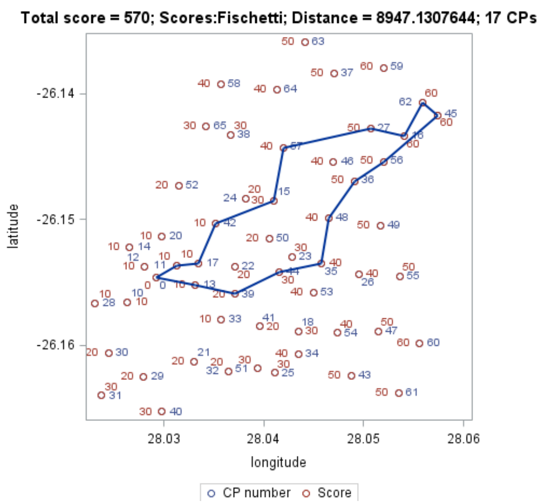
(b) 6 km



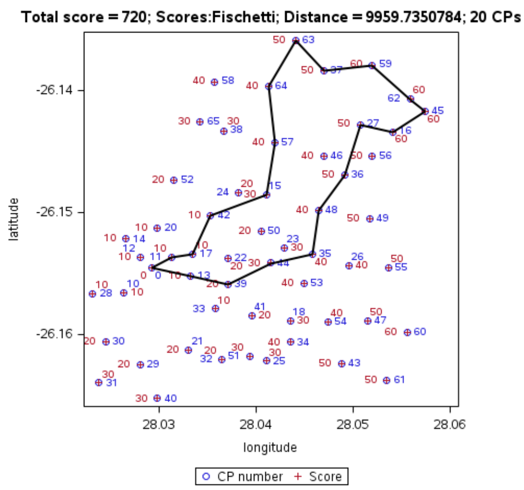
(c) 7 km



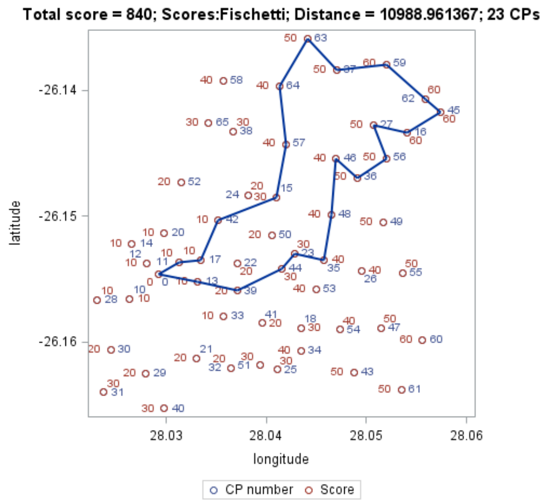
(d) 8 km



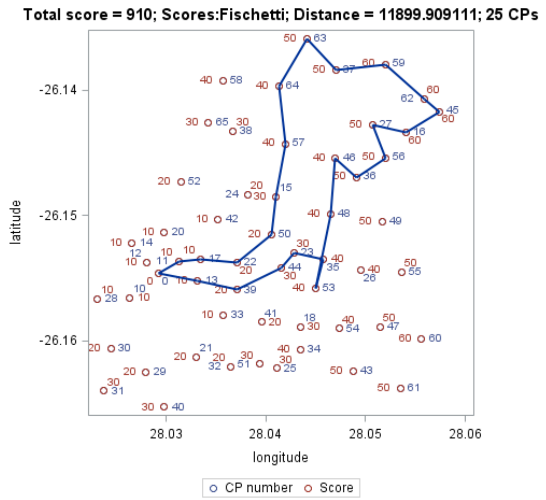
(e) 9 km



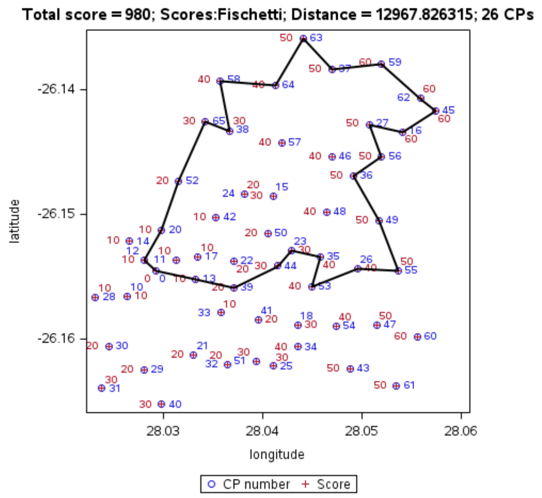
(f) 10 km



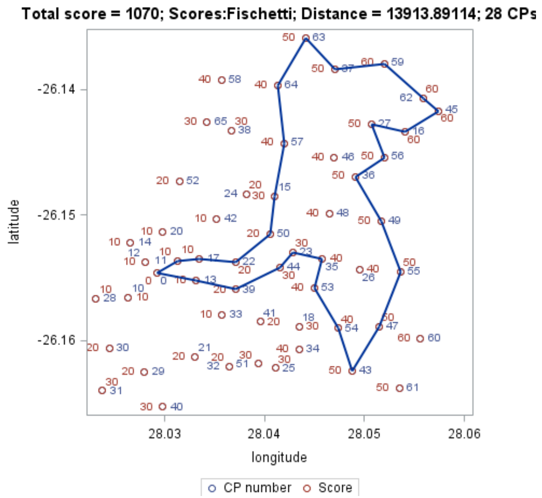
(g) 11 km



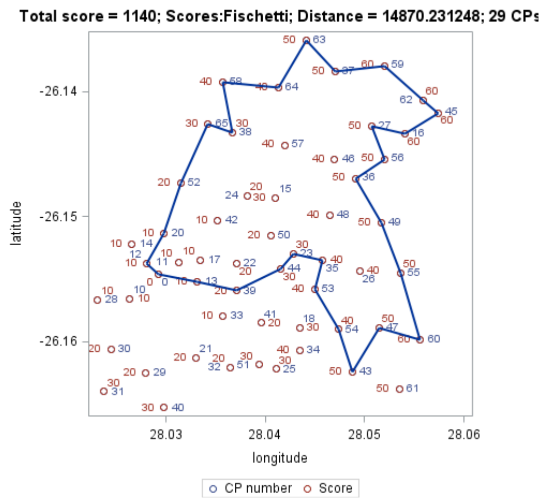
(h) 12 km



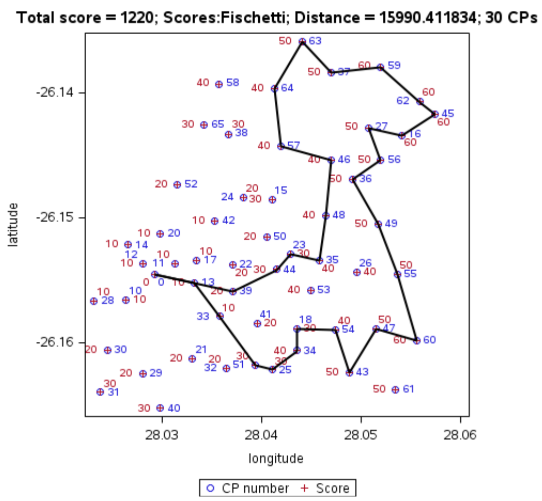
(i) 13 km



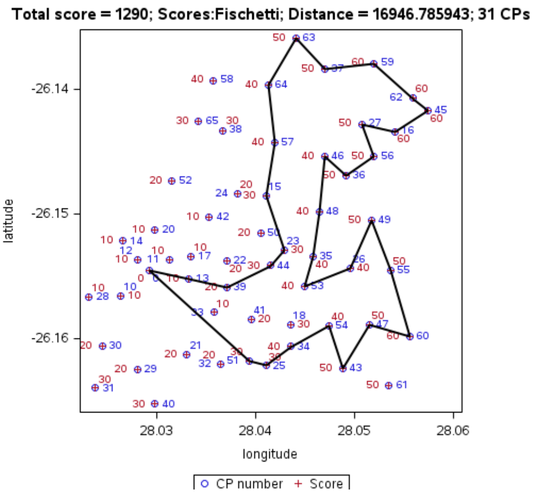
(j) 14 km



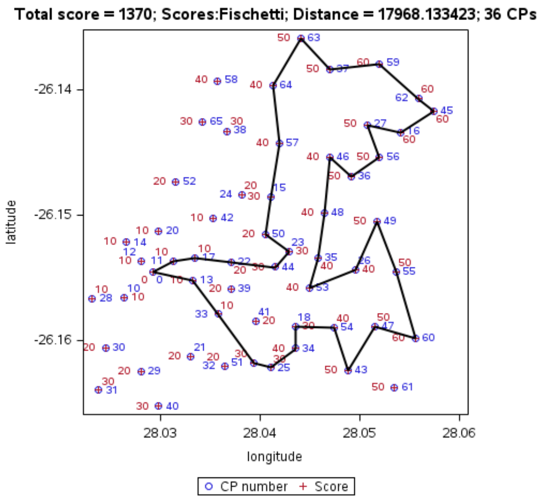
(k) 15 km



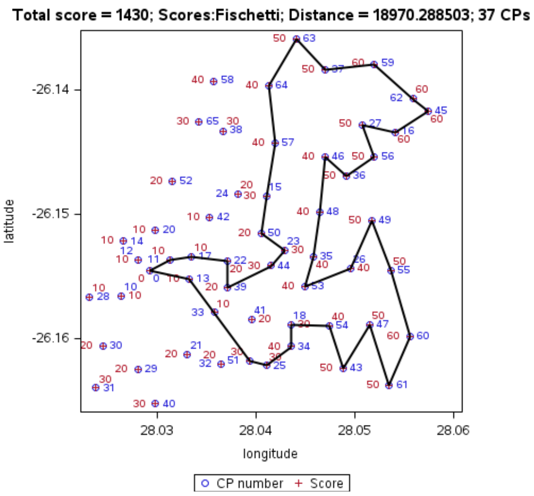
(l) 16 km



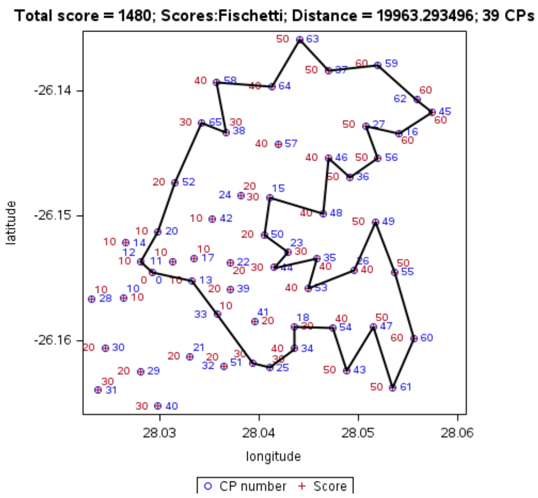
(m) 17 km



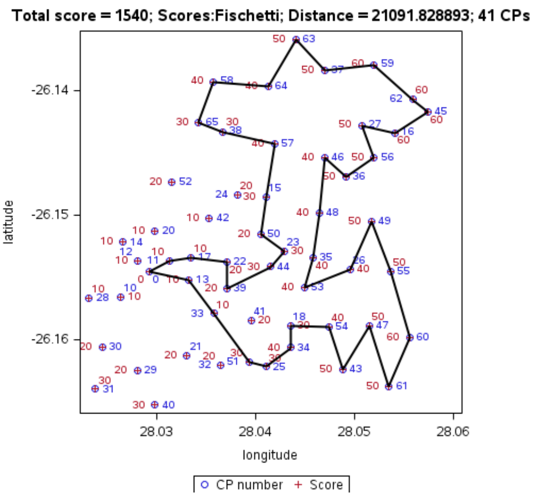
(n) 18 km



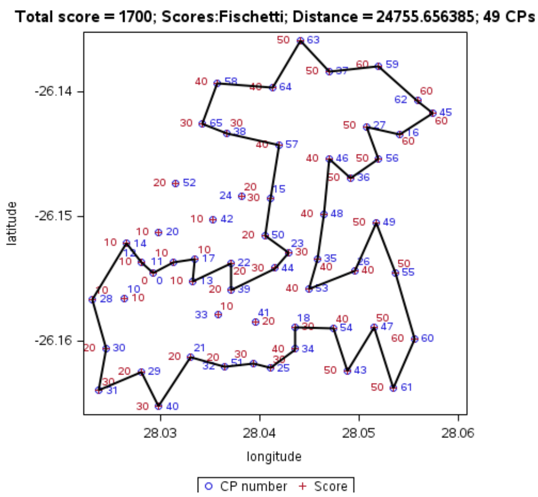
(o) 19 km



(p) 20 km



(q) 21,1 km



(r) 25 km

Appendix L

Visit frequency graphs: L.1 – L.26

L.1 SAME visits

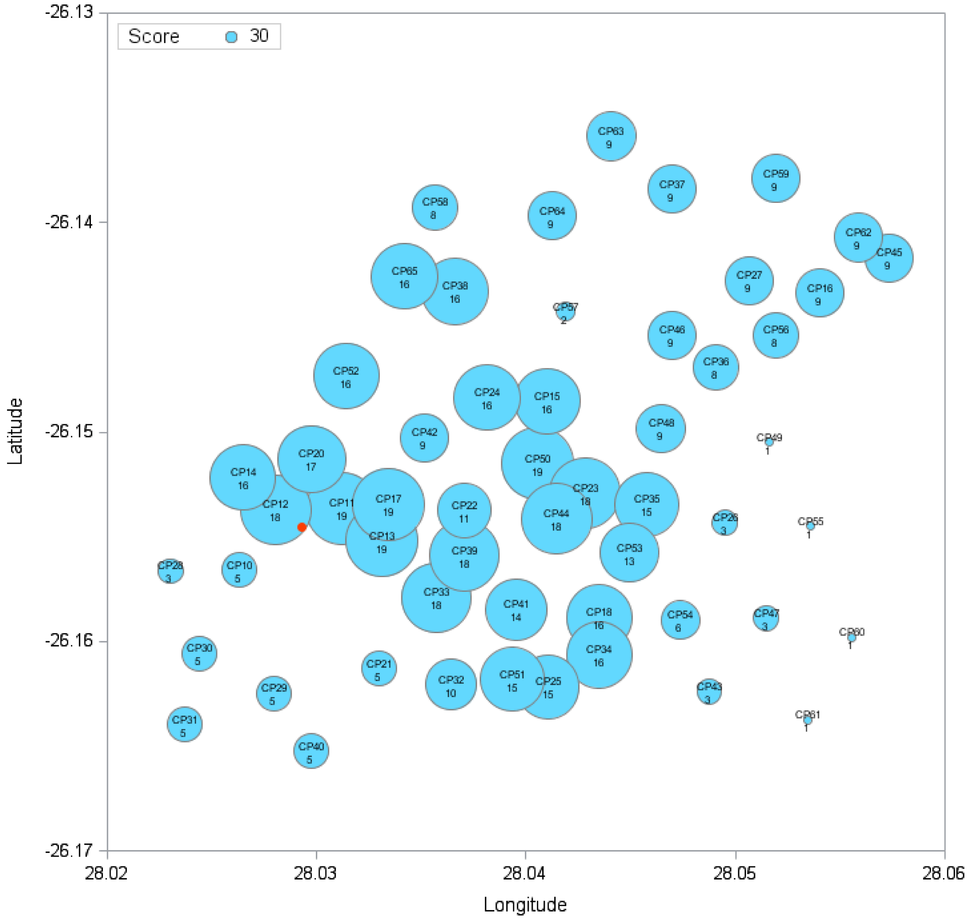


Figure L.1: Visit frequencies resulting from SAME

L.2 SAME_I visits

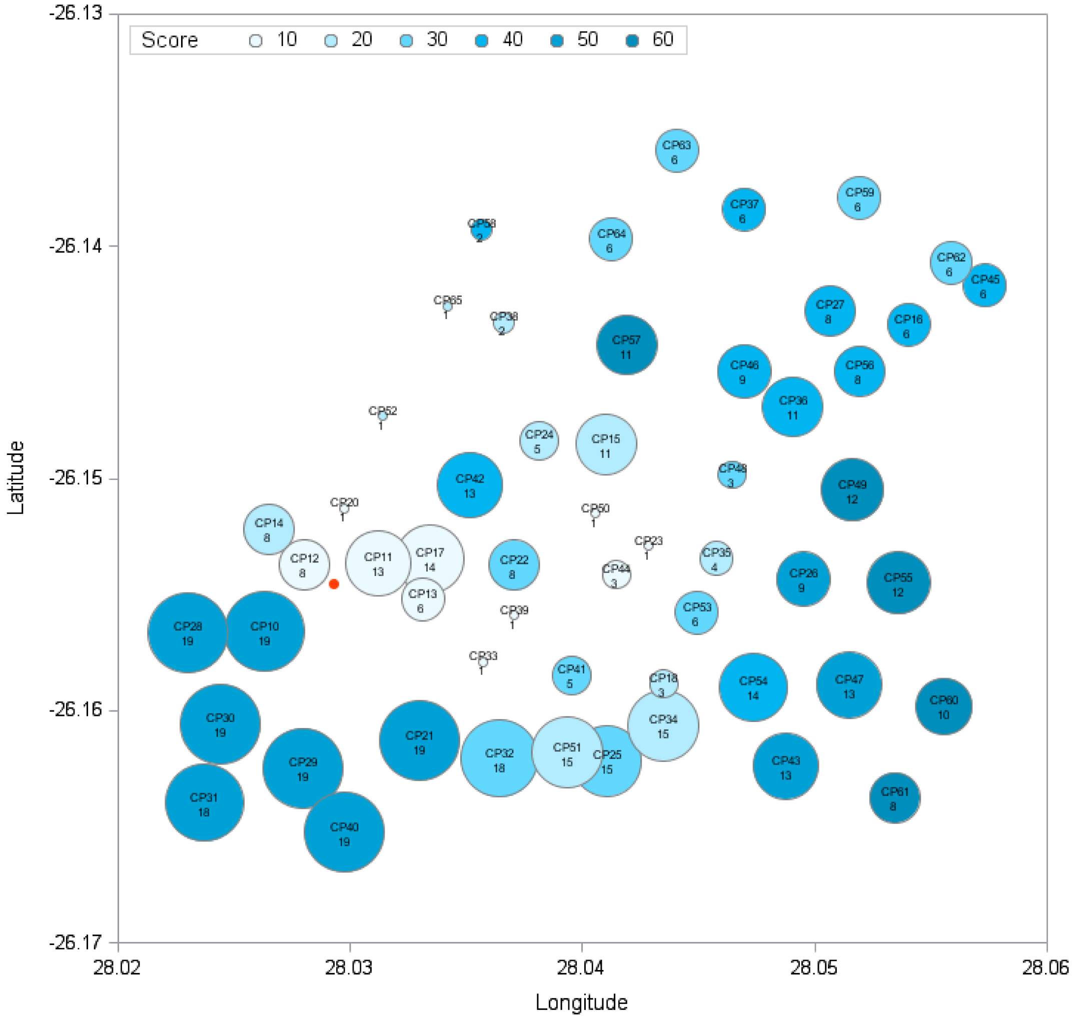


Figure L.2: Visit frequencies resulting from SAME_I

L.3 SAME_II visits

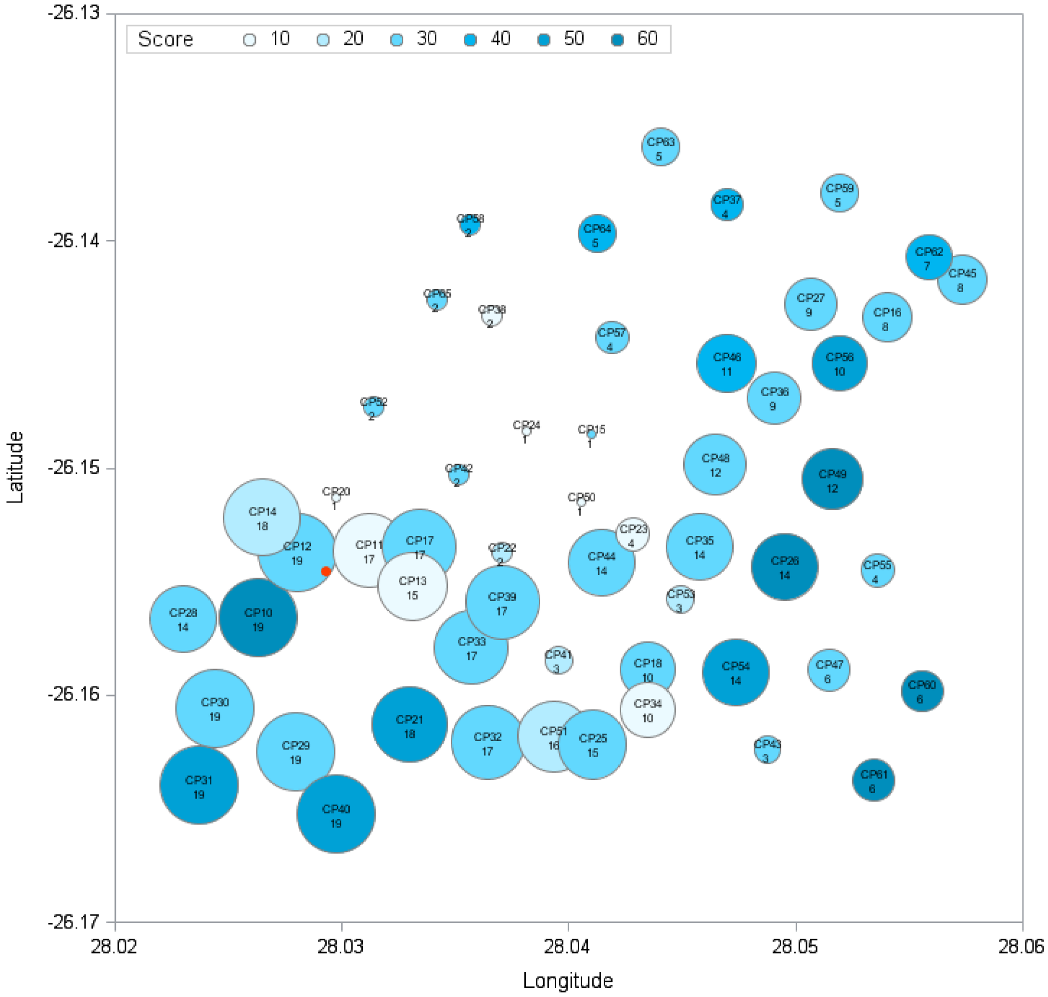


Figure L.3: Visit frequencies resulting from SAME_II

L.4 SAME_III visits

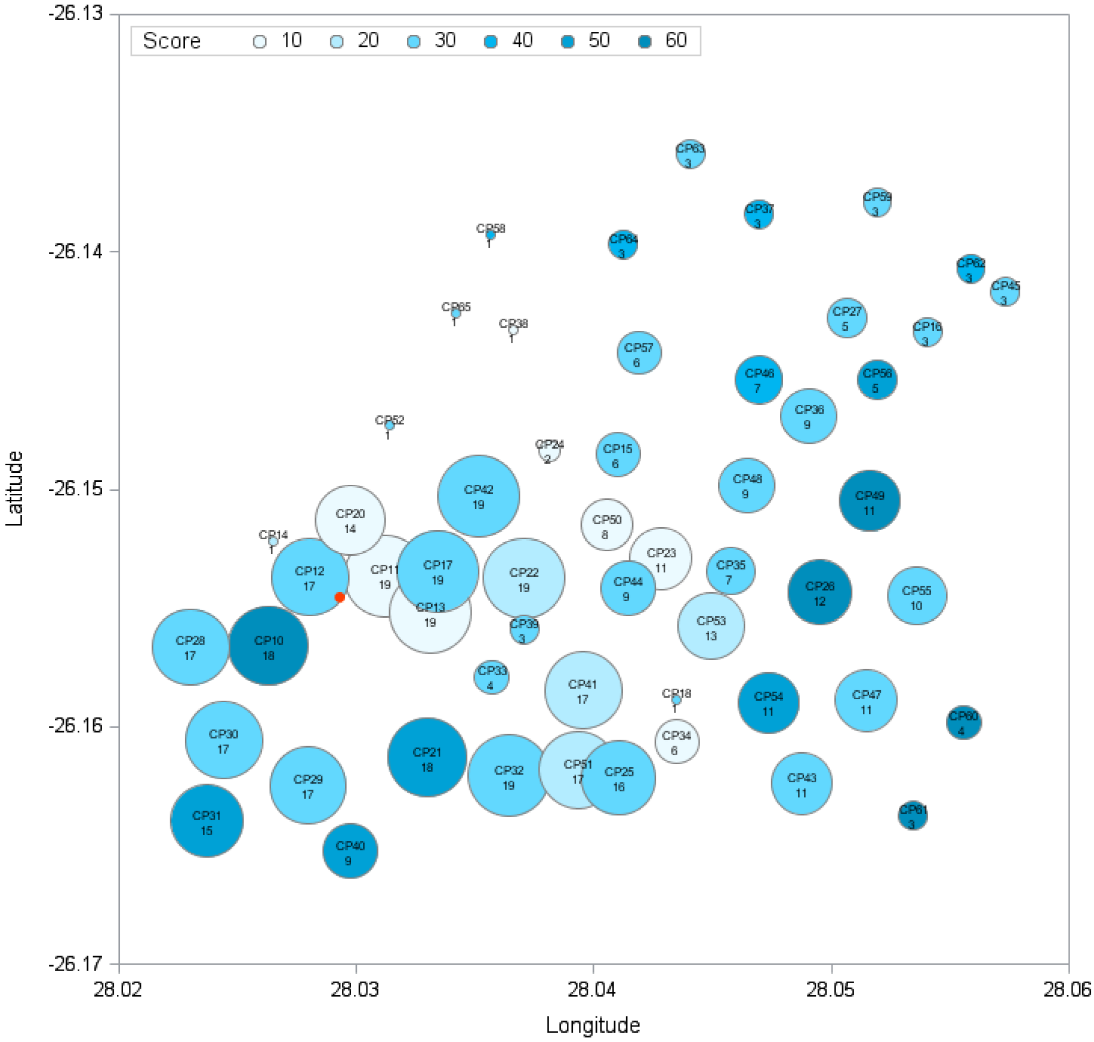


Figure L.4: Visit frequencies resulting from SAME_III

L.5 TSP visits

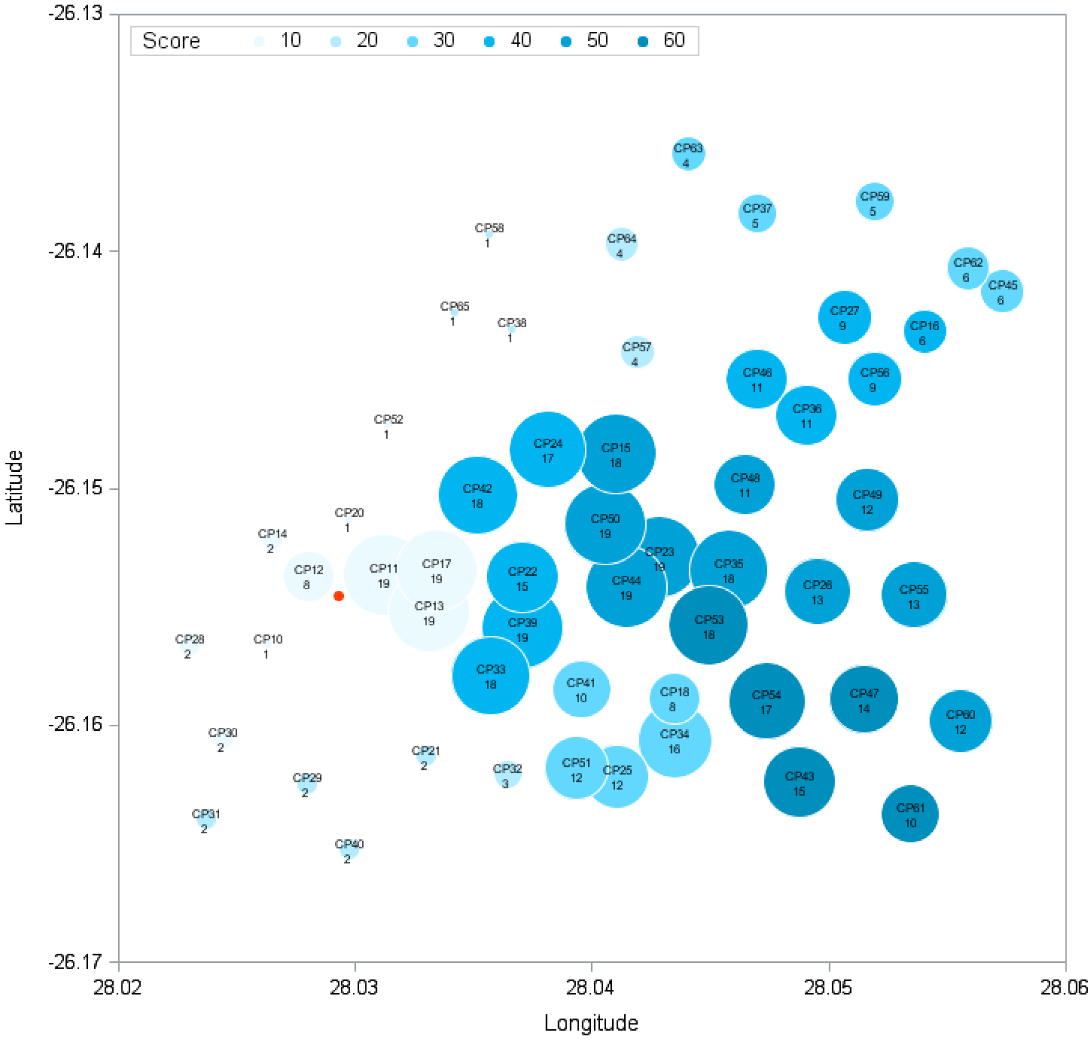


Figure L.5: Visit frequencies resulting from TSP

L.6 RANDOM visits

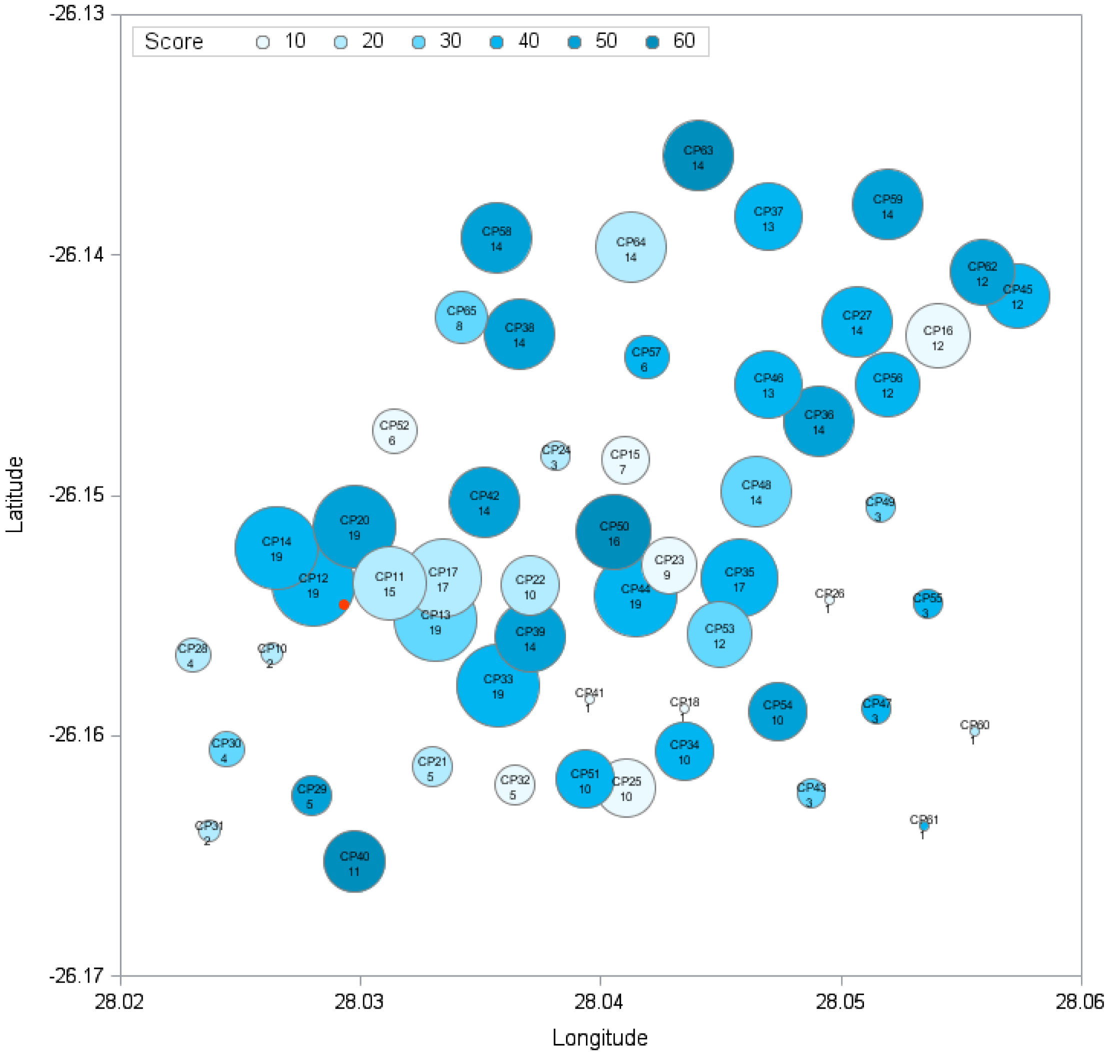


Figure L.6: Visit frequencies resulting from RANDOM

L.7 EXIST visits

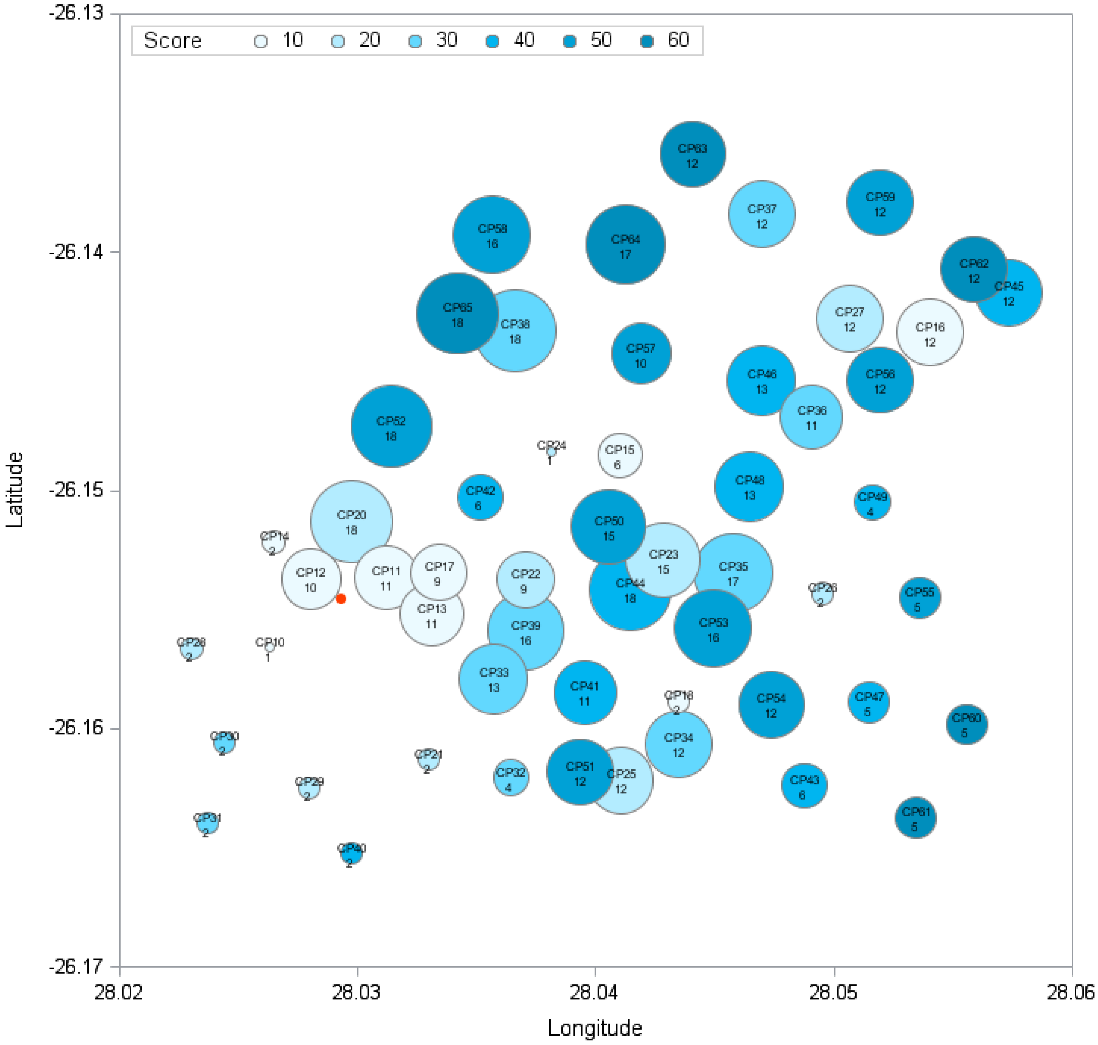


Figure L.7: Visit frequencies resulting from EXISTING

L.8 Geo100G0L visits

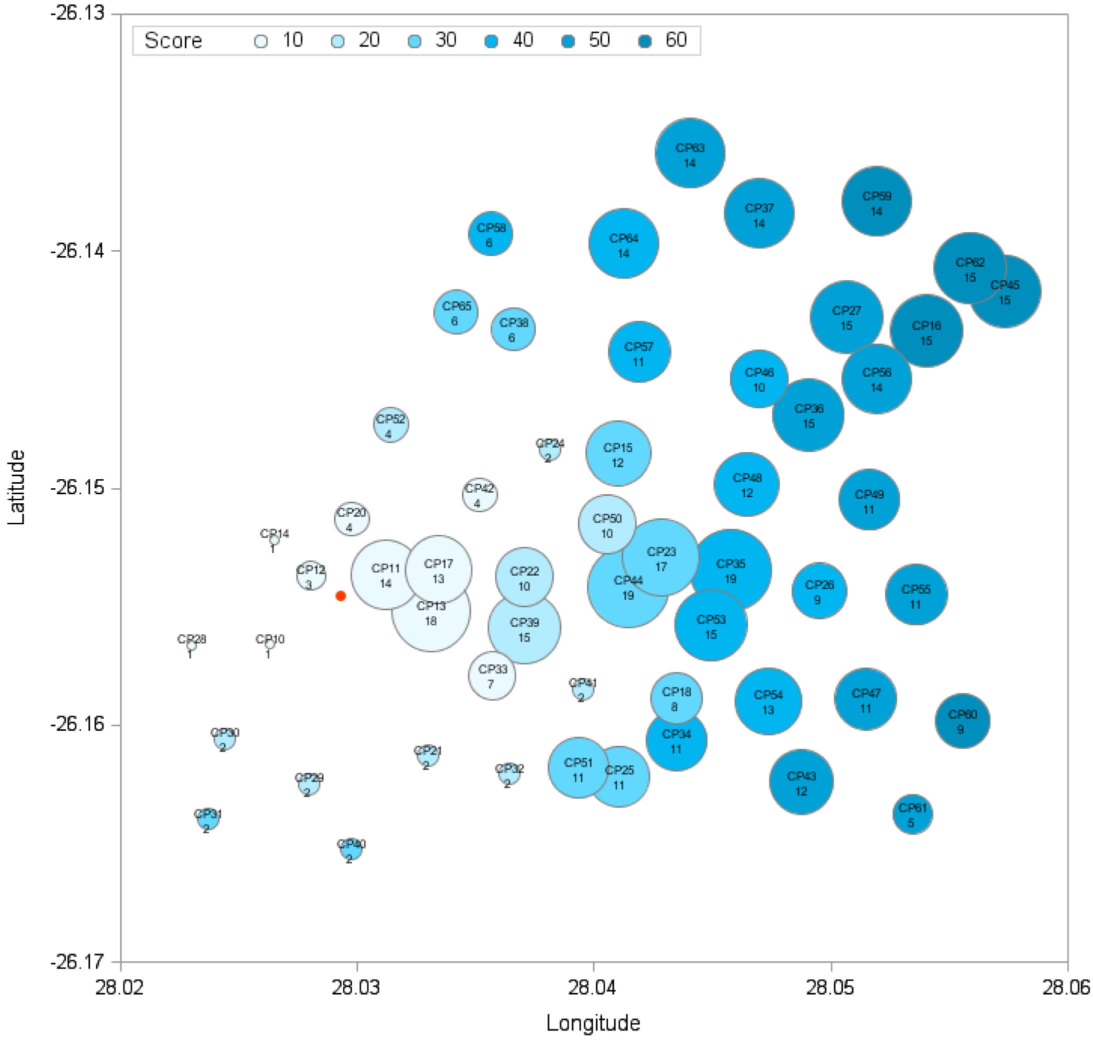


Figure L.8: Visit frequencies resulting from Geo100G0L

L.9 Geo80G20L visits

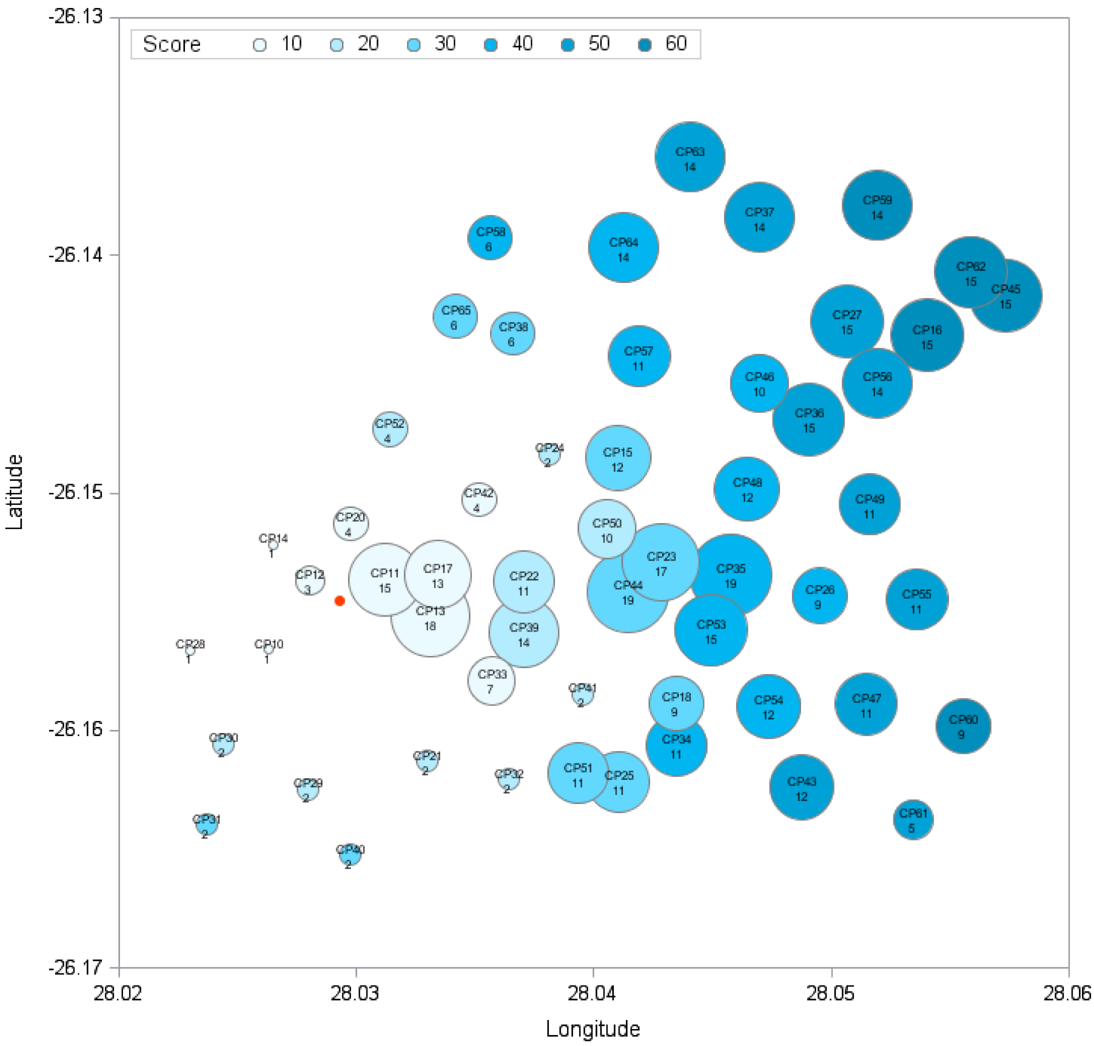


Figure L.9: Visit frequencies resulting from Geo80G20L

L.10 Geo50G50L visits

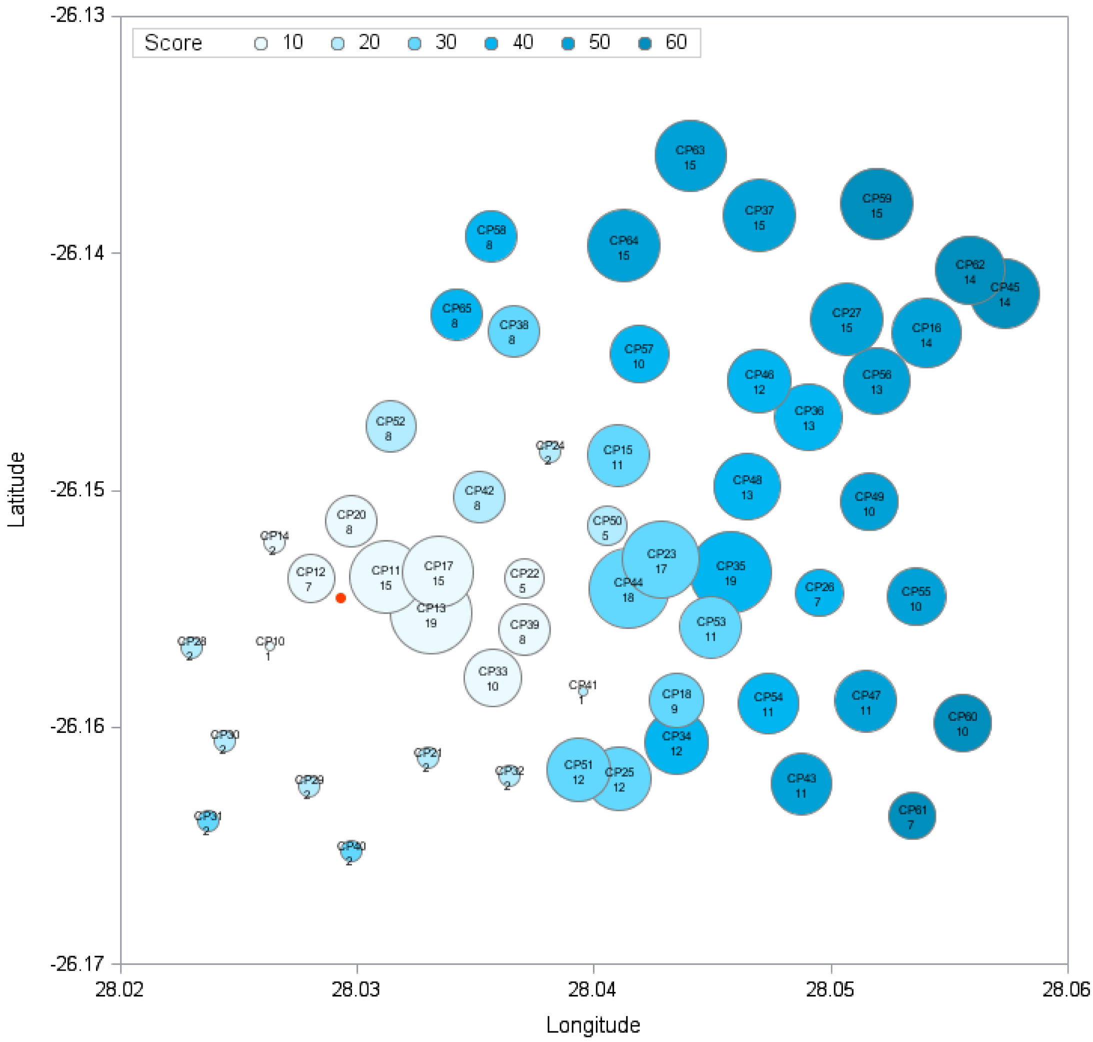


Figure L.10: Visit frequencies resulting from Geo50G50L

L.11 Geo20G80L visits

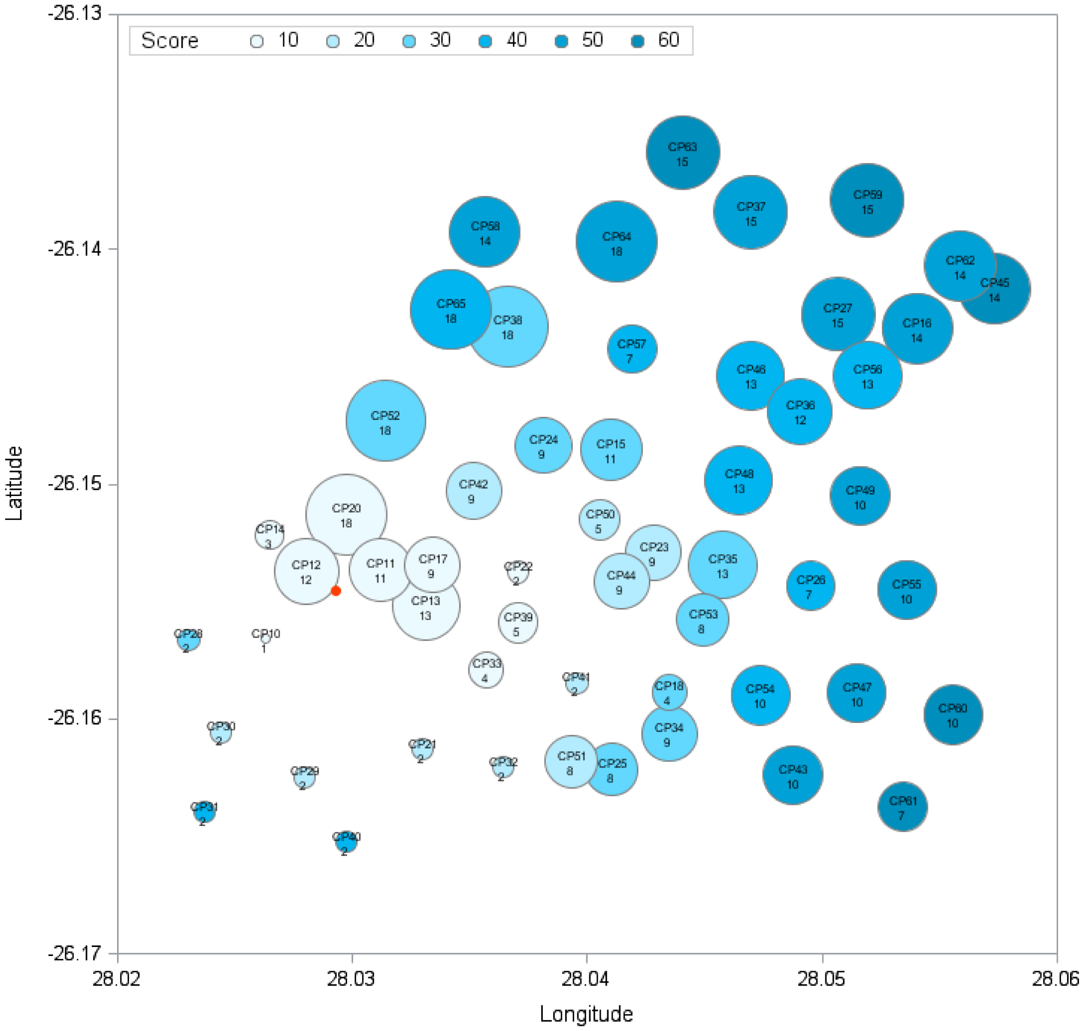


Figure L.11: Visit frequencies resulting from Geo20G80L

L.12 Geo0G100L visits

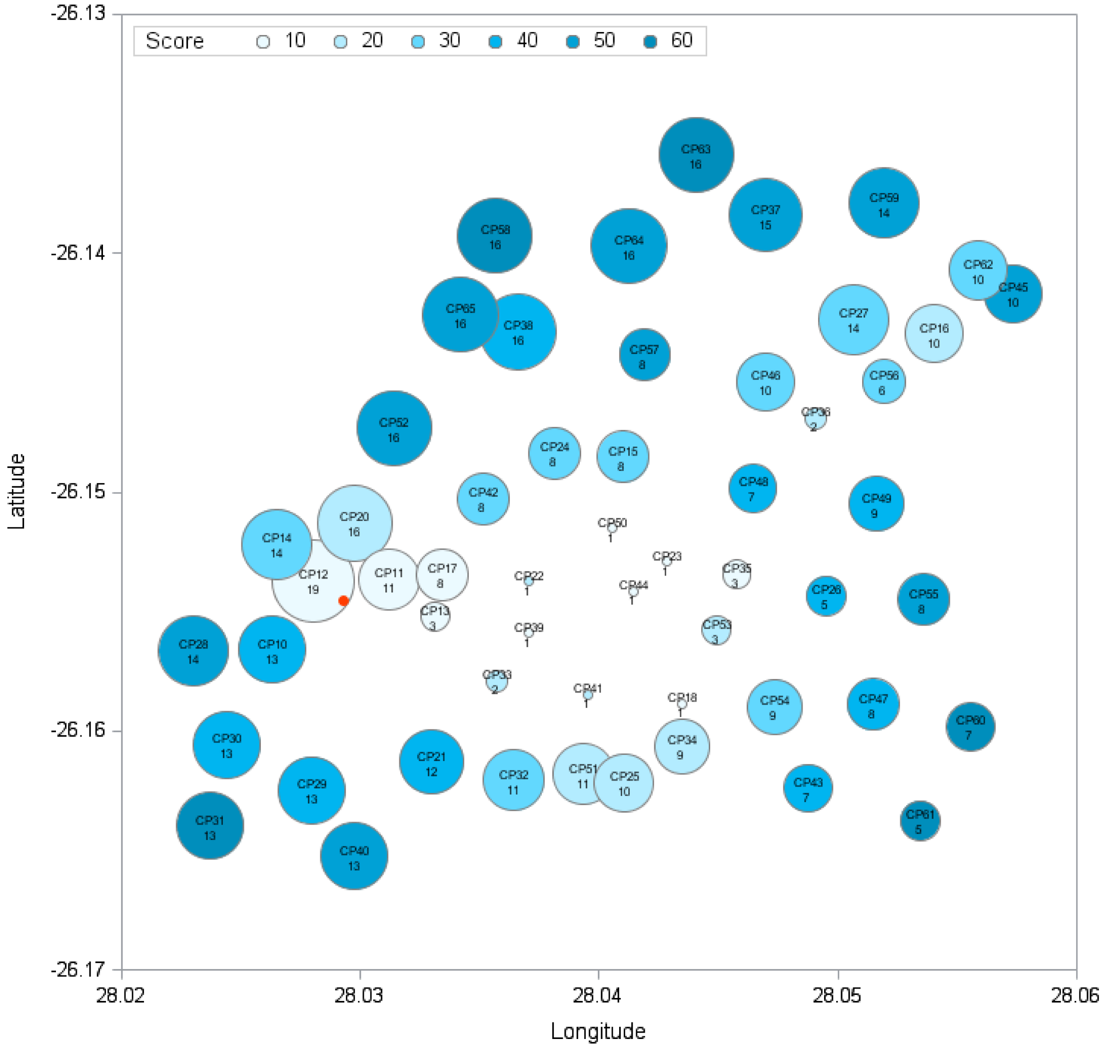


Figure L.12: Visit frequencies resulting from Geo0G100L

L.13 Road100G0L visits

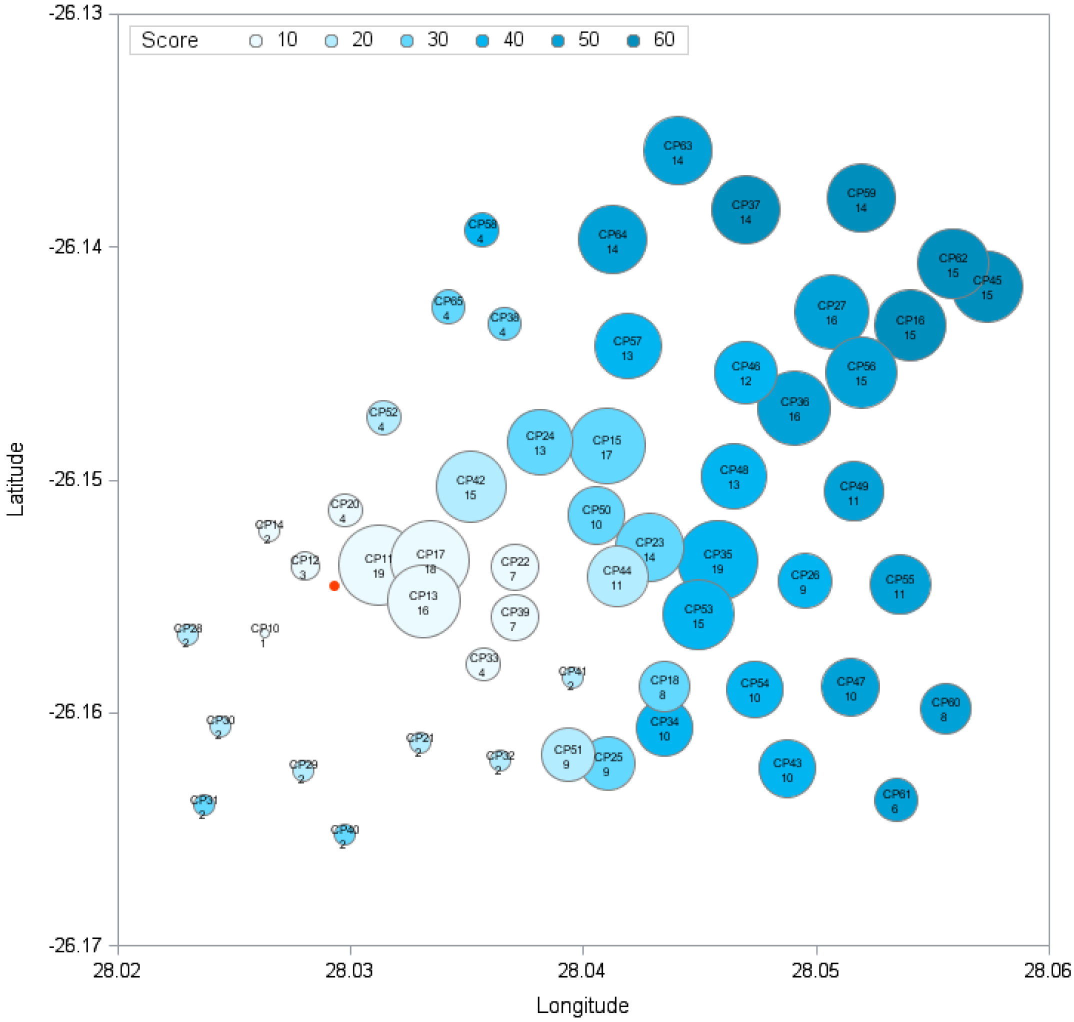


Figure L.13: Visit frequencies resulting from Road100G0L

L.14 Road80G20L visits

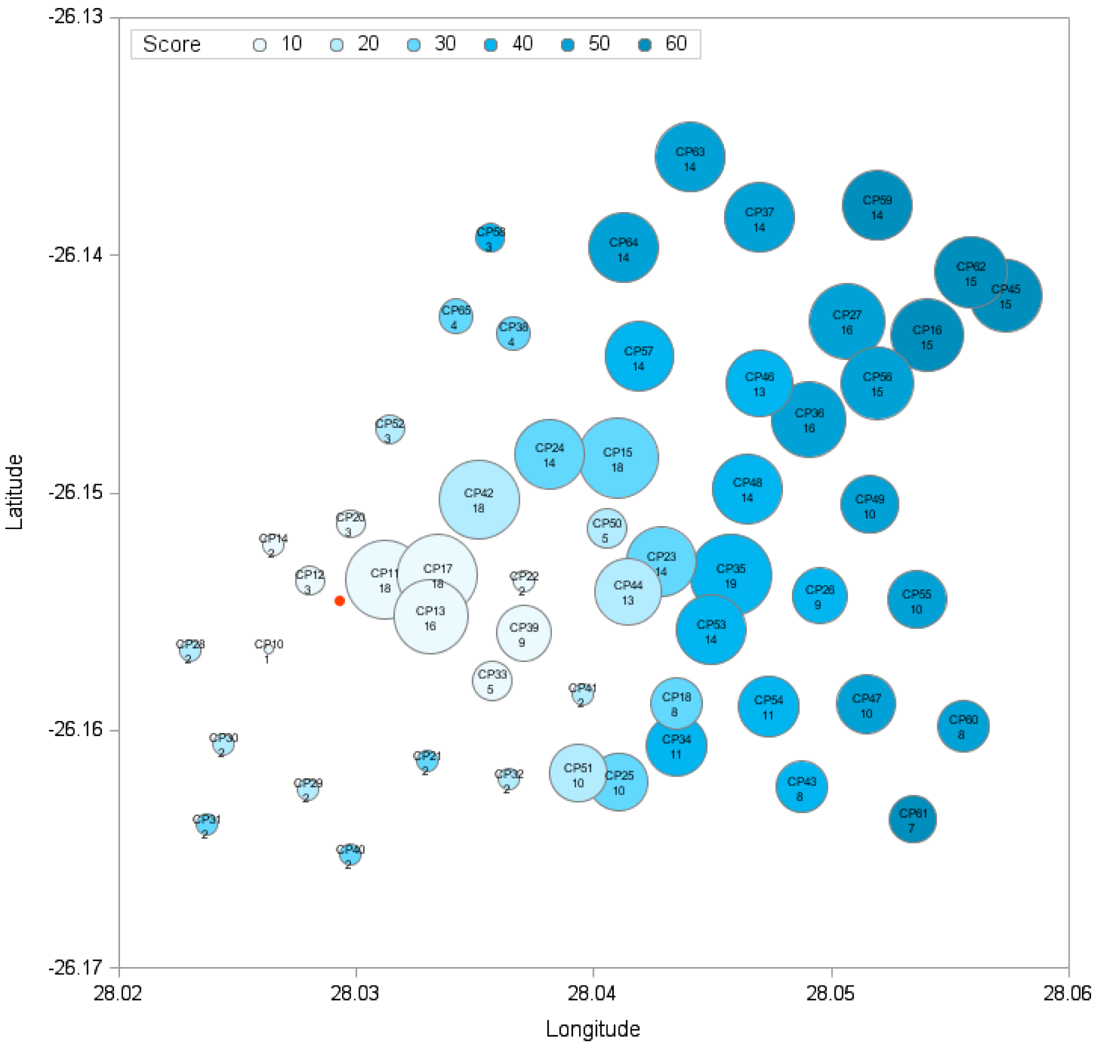


Figure L.14: Visit frequencies resulting from Road80G20L

L.15 Road50G50L visits

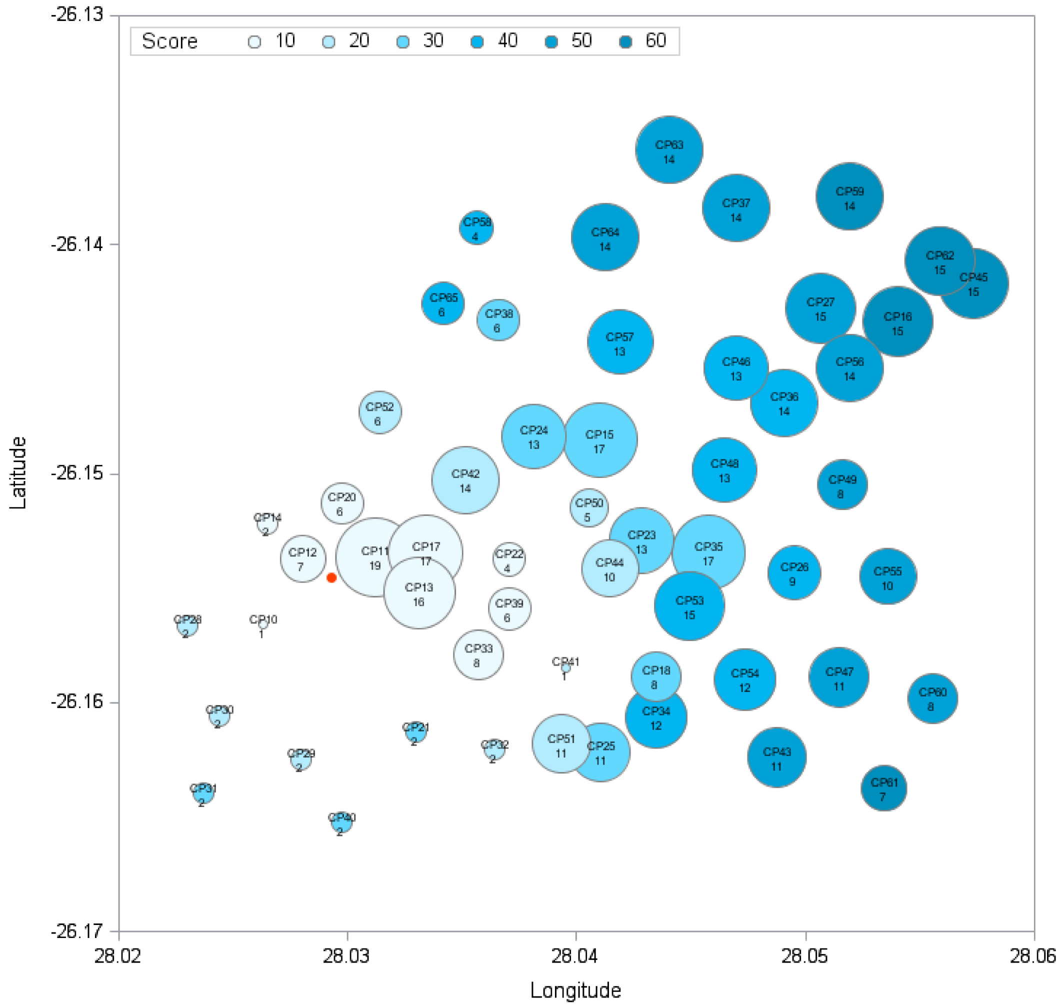


Figure L.15: Visit frequencies resulting from Road50G50L

L.16 Road20G80L visits

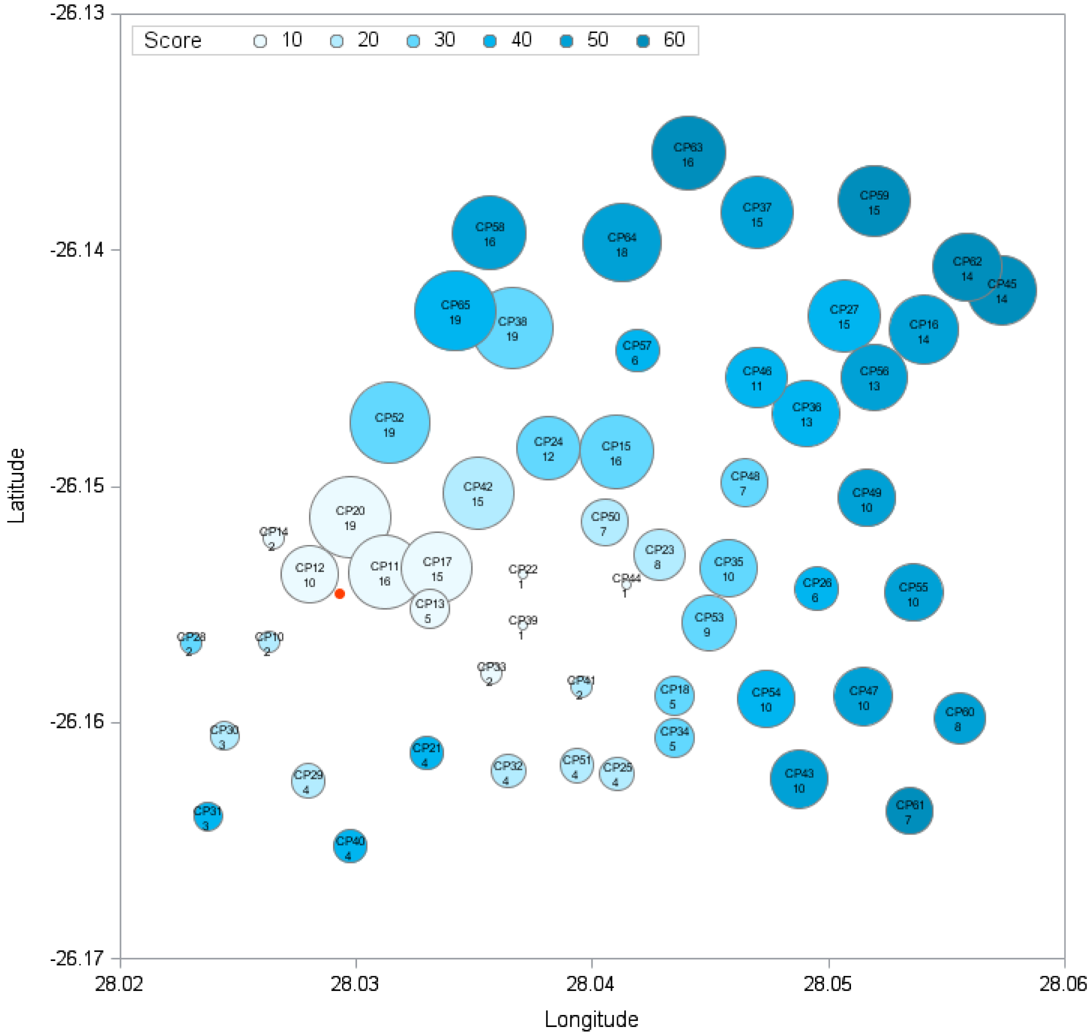


Figure L.16: Visit frequencies resulting from Road20G80L

L.17 Road0G100L visits

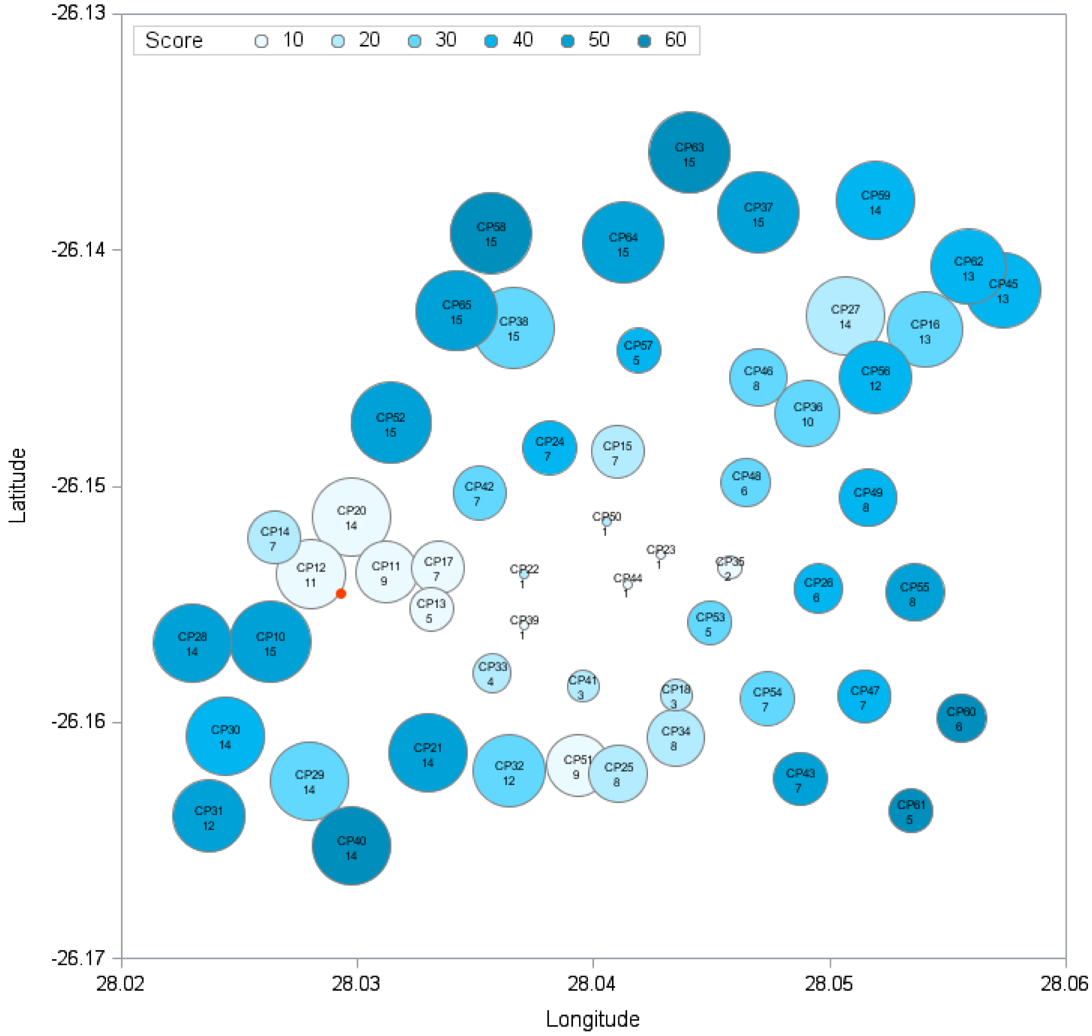


Figure L.17: Visit frequencies resulting from Road0G100L

L.18 Alt_adj100G0L visits

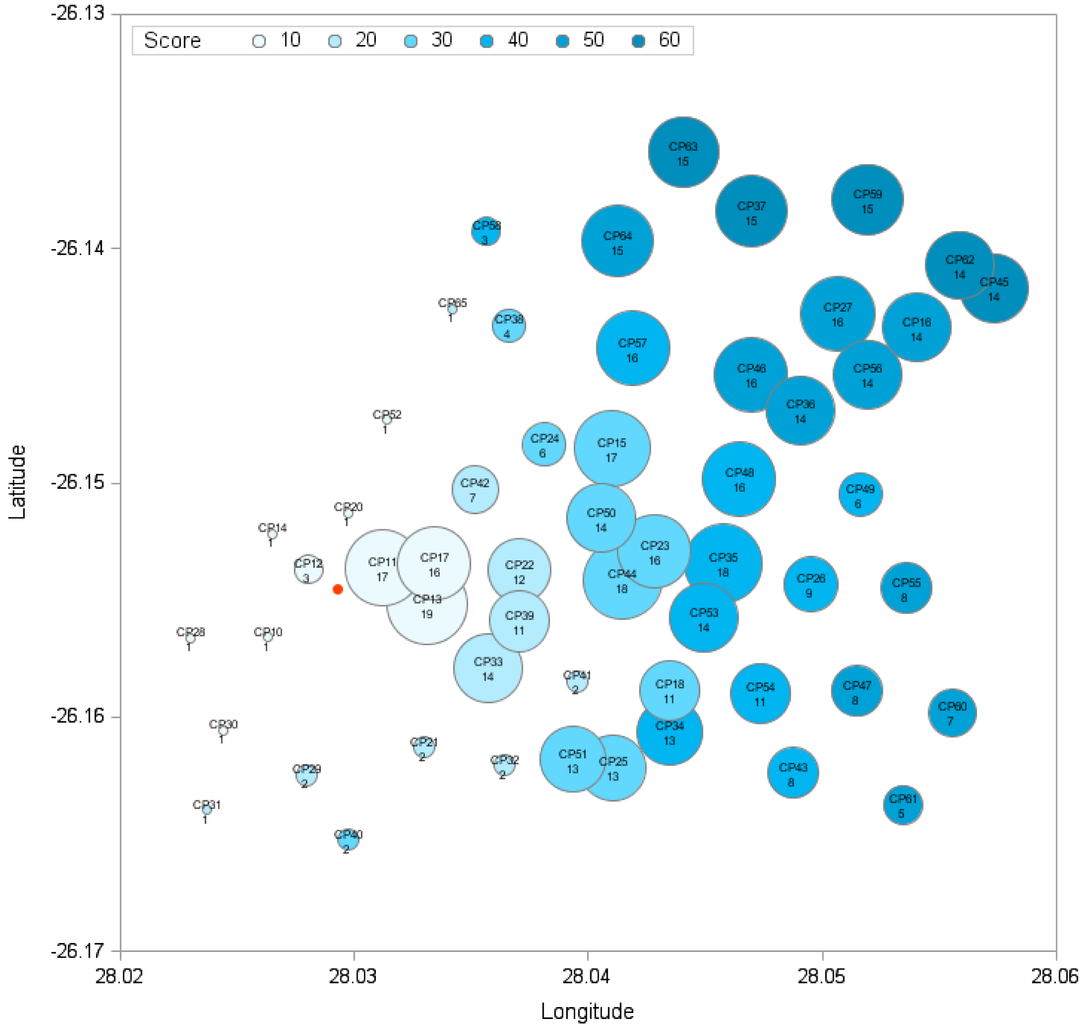


Figure L.18: Visit frequencies resulting from Alt_adj100G0L

L.19 Alt_adj80G20L visits

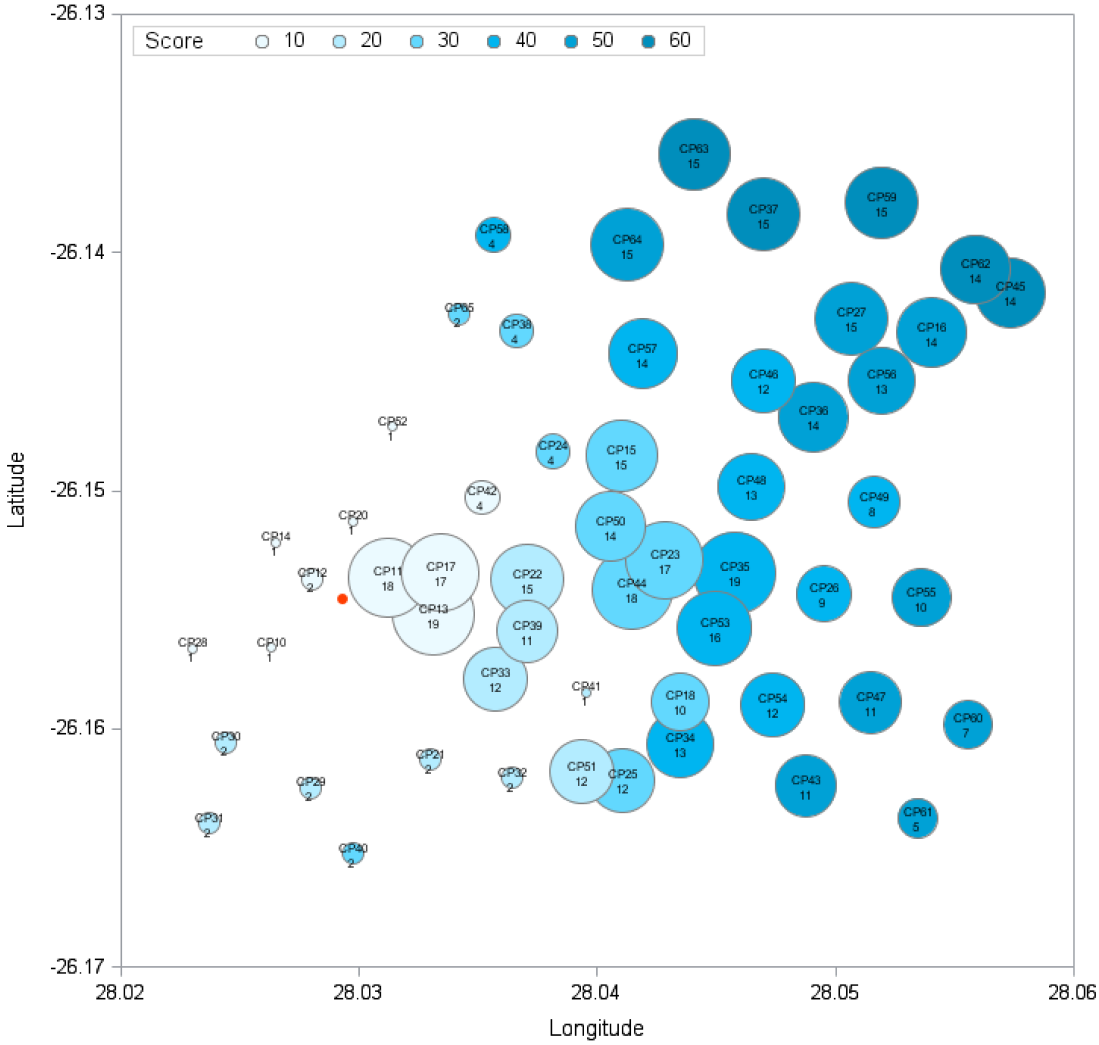


Figure L.19: Visit frequencies resulting from Alt_adj80G20L

L.20 Alt_adj50G50L visits

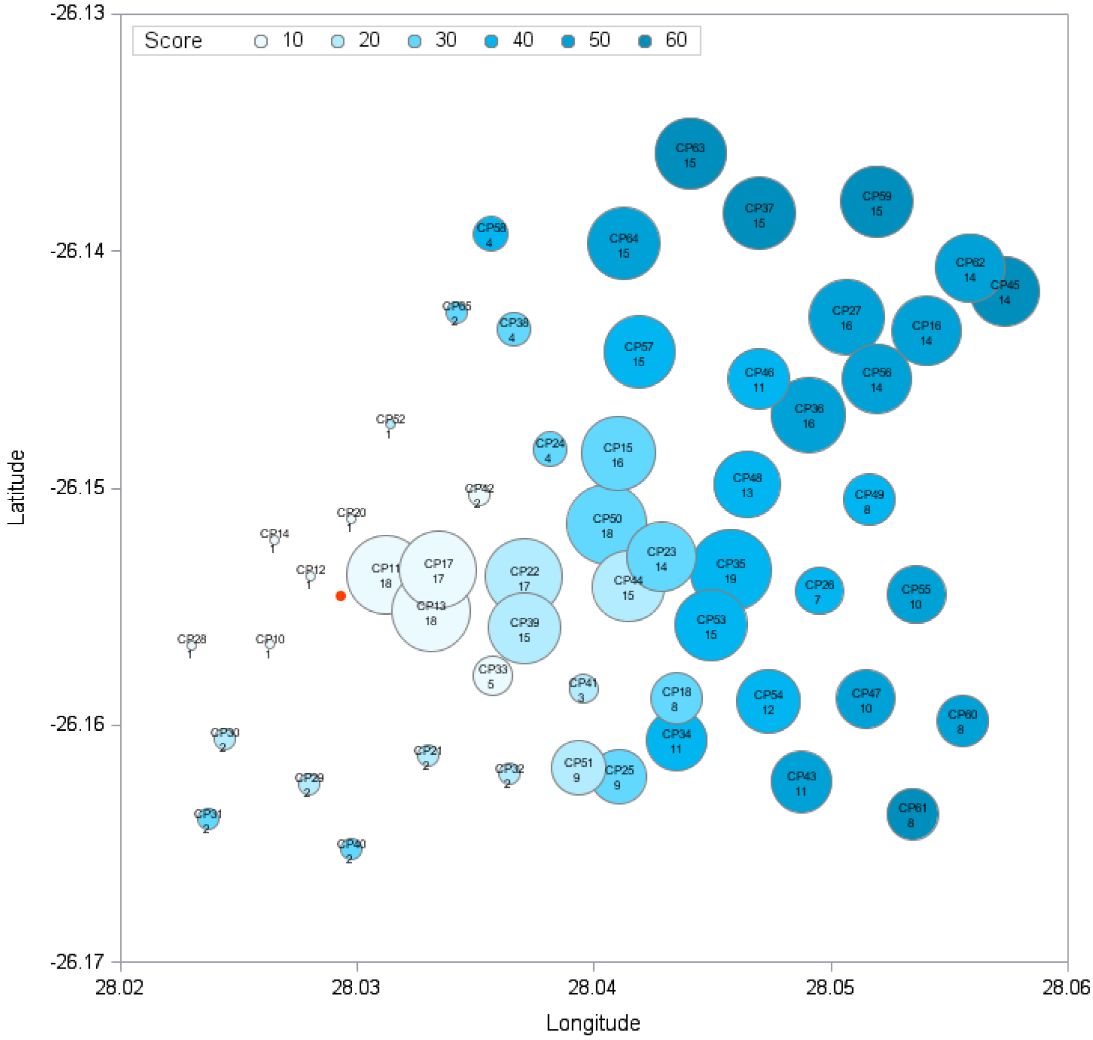


Figure L.20: Visit frequencies resulting from Alt_adj50G50L

L.21 Alt_adj20G80L visits

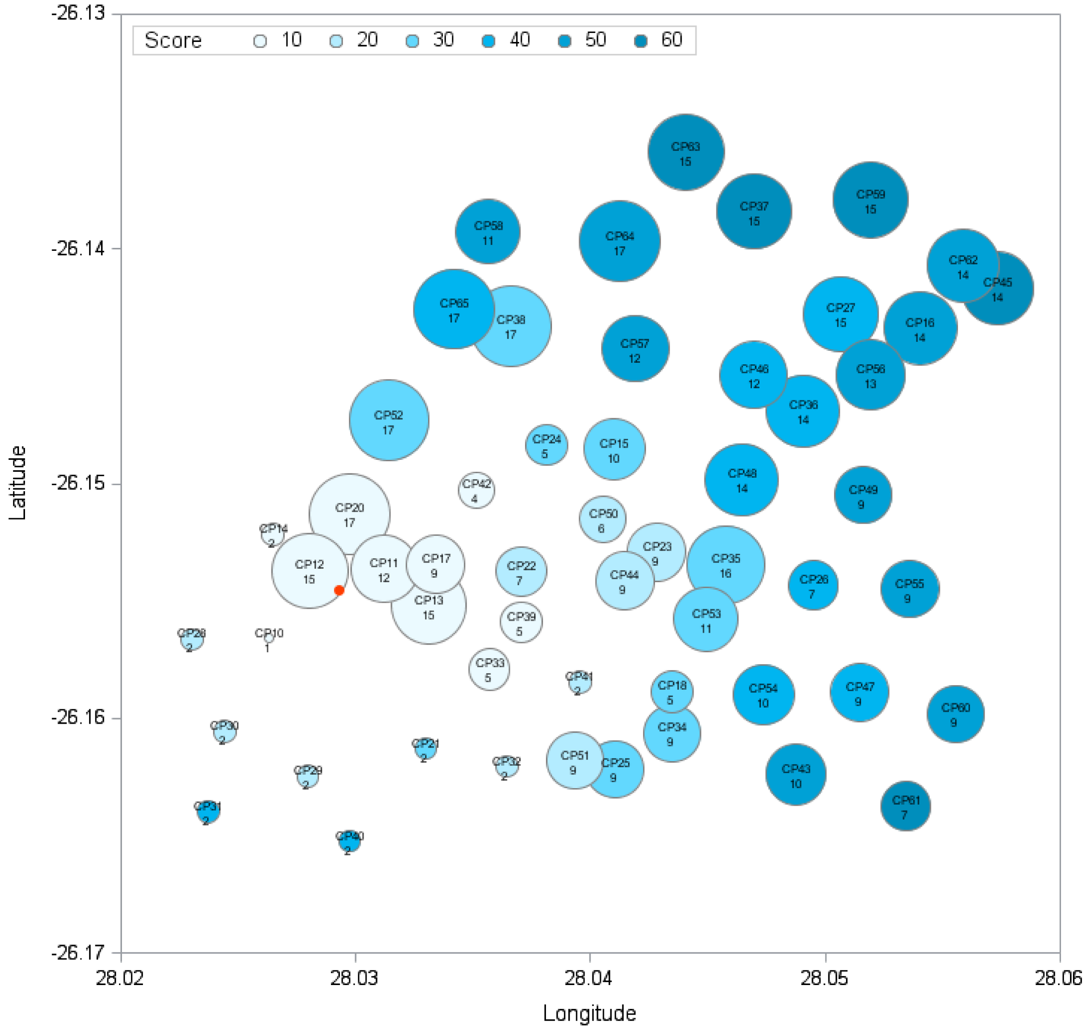


Figure L.21: Visit frequencies resulting from Alt_adj20G80L

L.22 Alt_adj0G100L visits

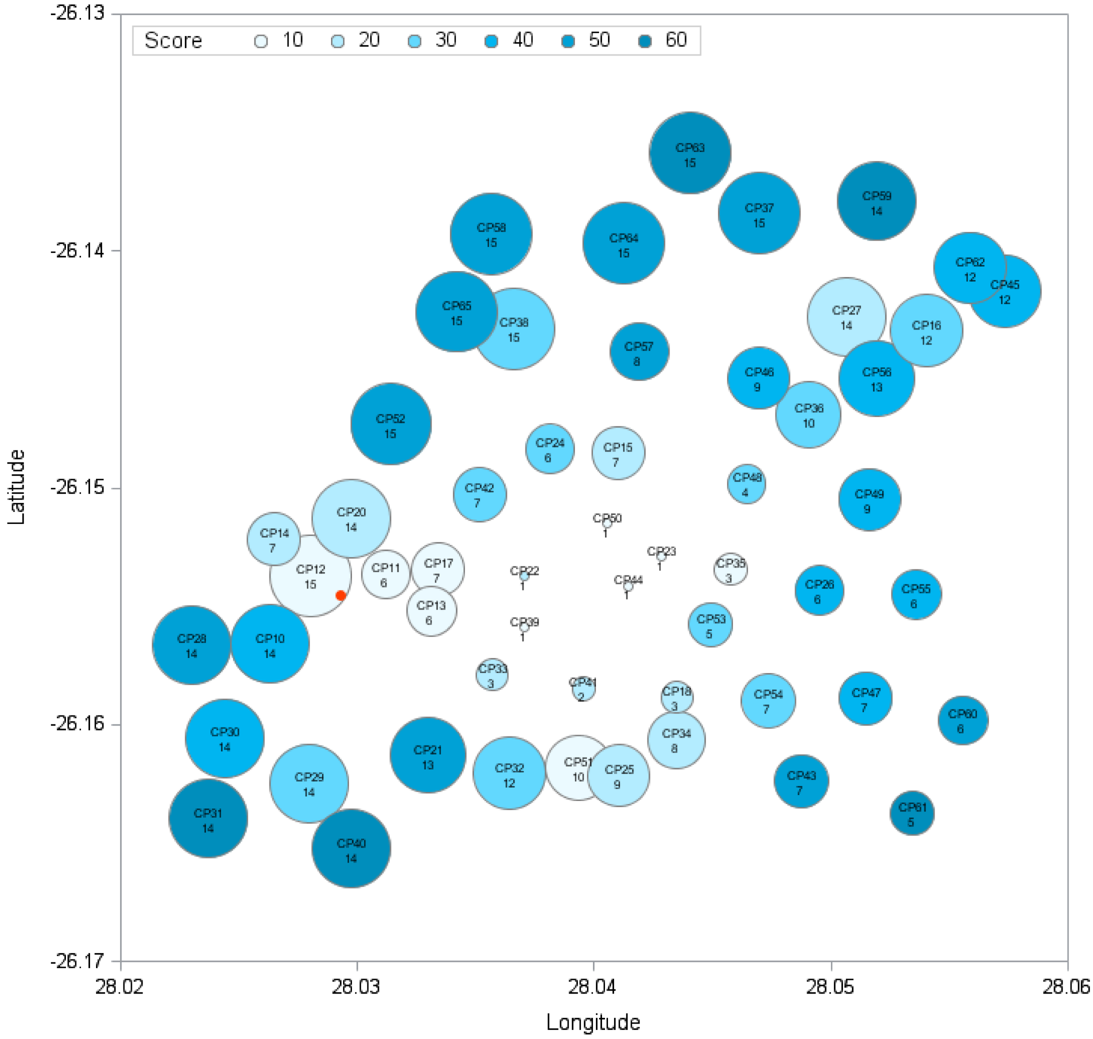


Figure L.22: Visit frequencies resulting from Alt_adj0G100L

L.23 TS0Si100Sii visits

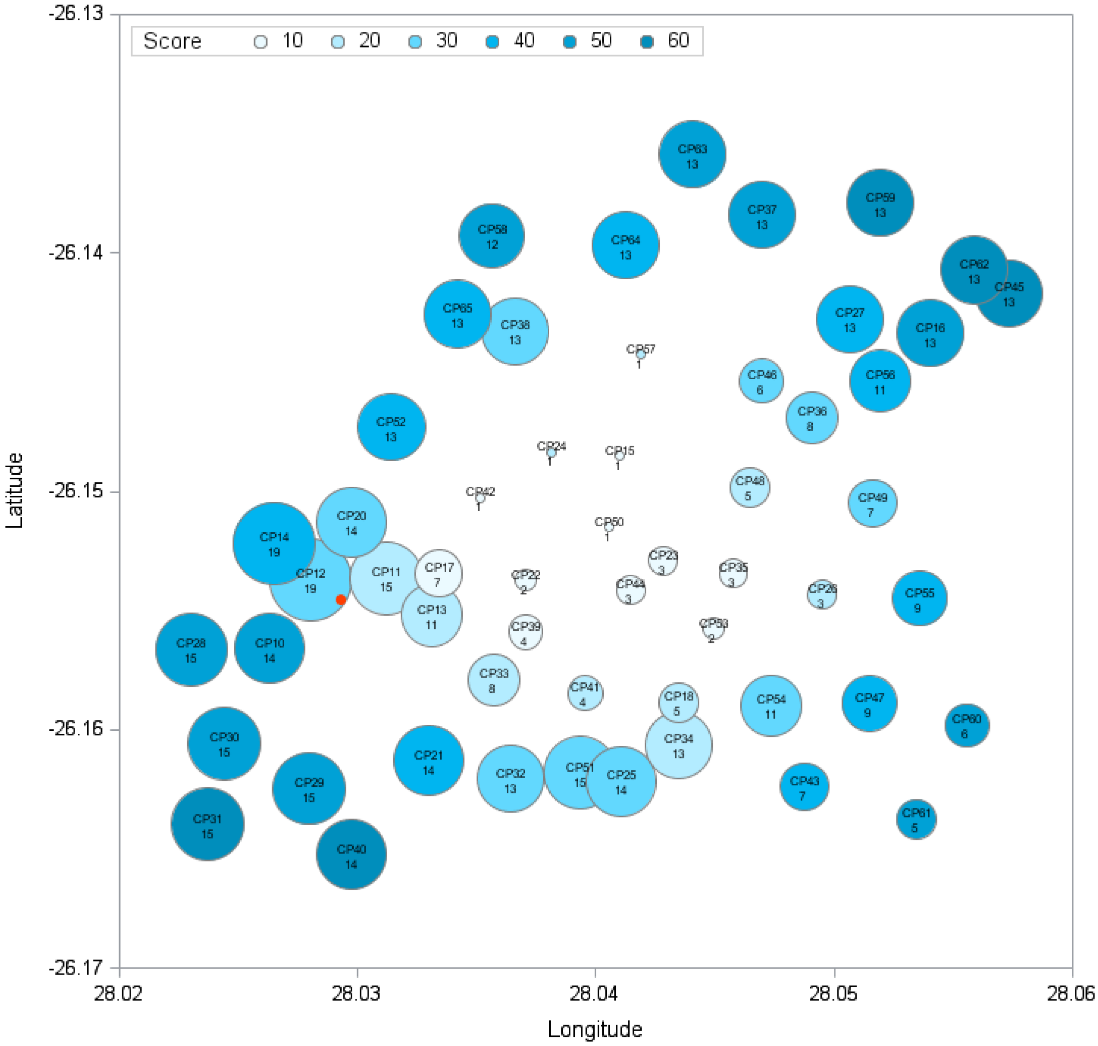


Figure L.23: Visit frequencies resulting from TS0Si100Sii

L.24 TS50Si50Sii visits

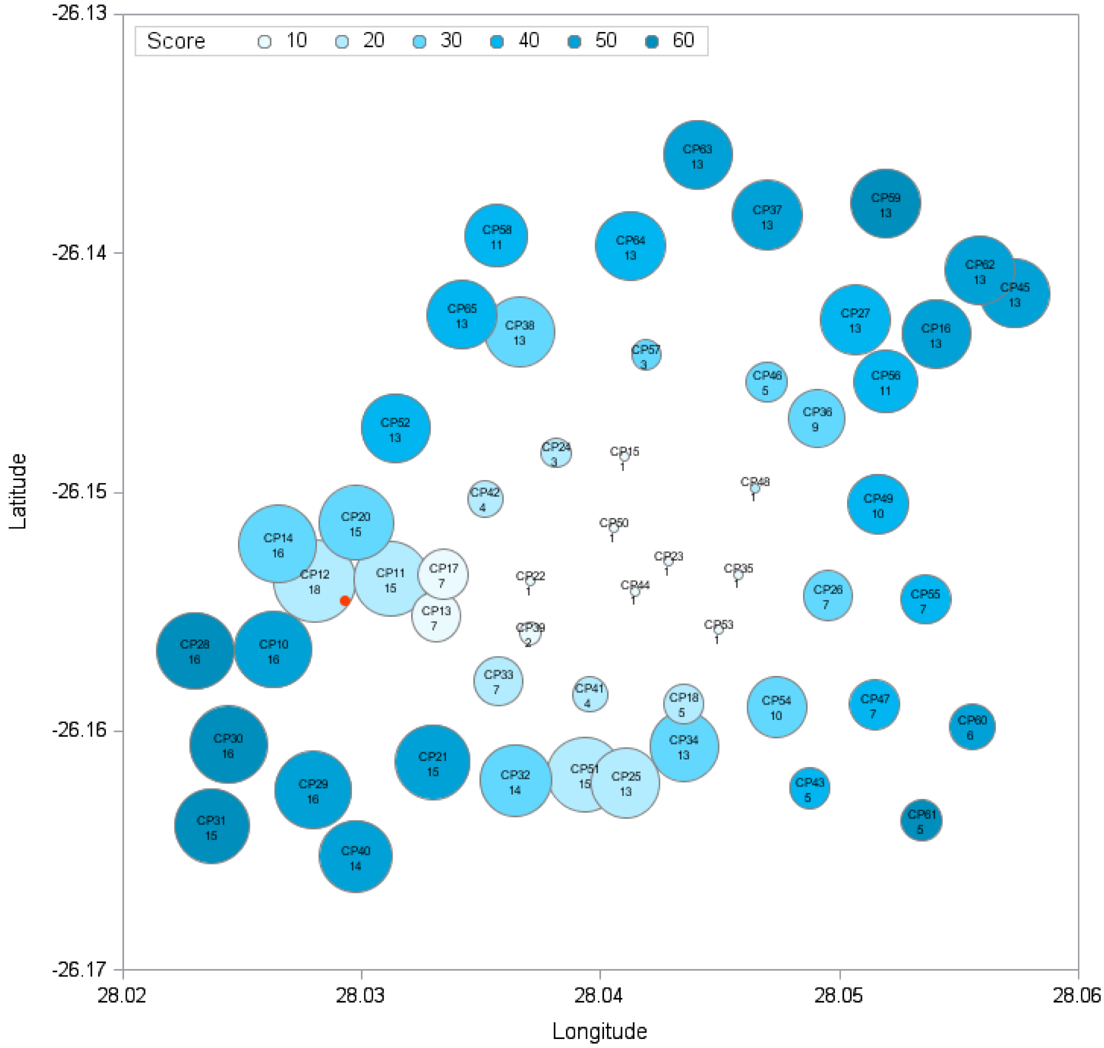


Figure L.24: Visit frequencies resulting from TS50Si50Sii

L.25 TSIG100si0sii visits

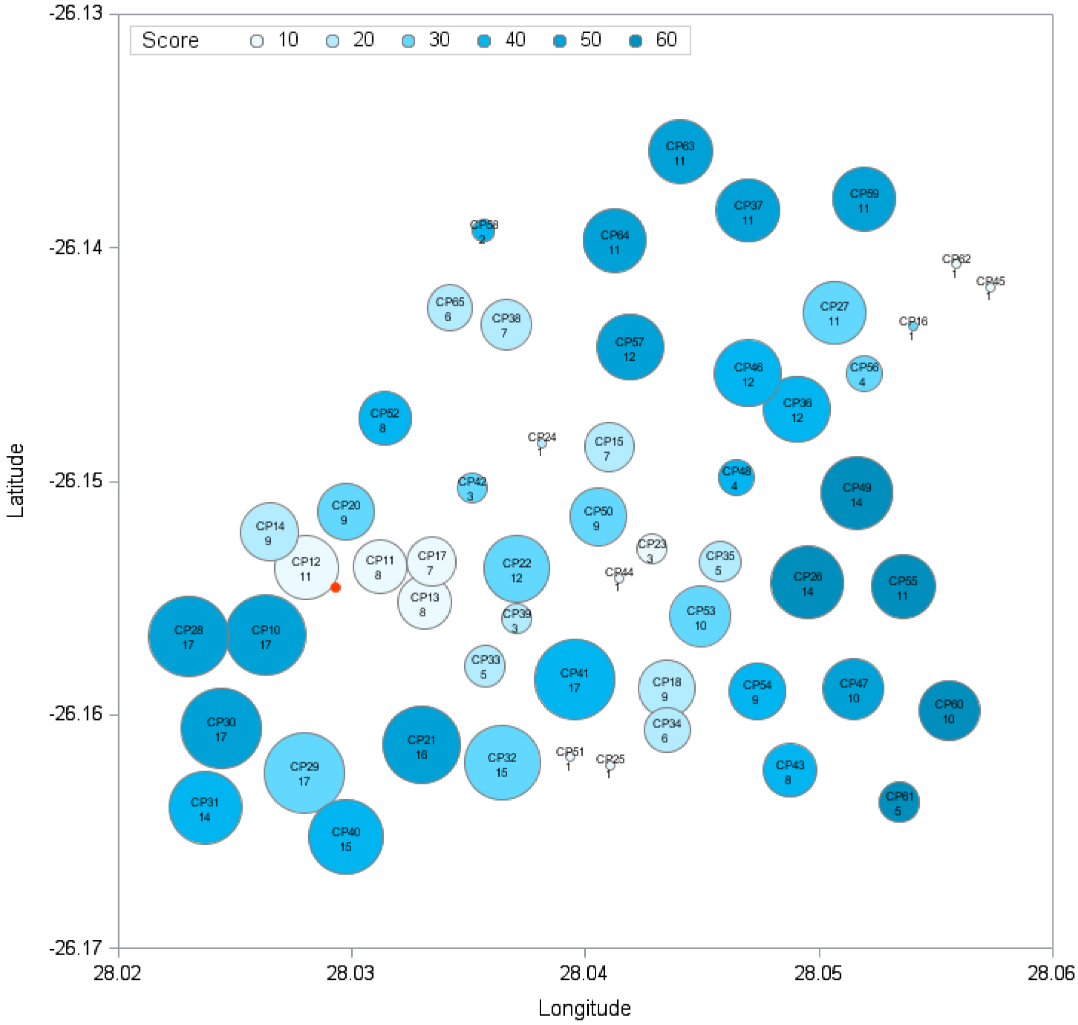


Figure L.25: Visit frequencies resulting from TSIG100si0sii

L.26 FISCHETTI visits

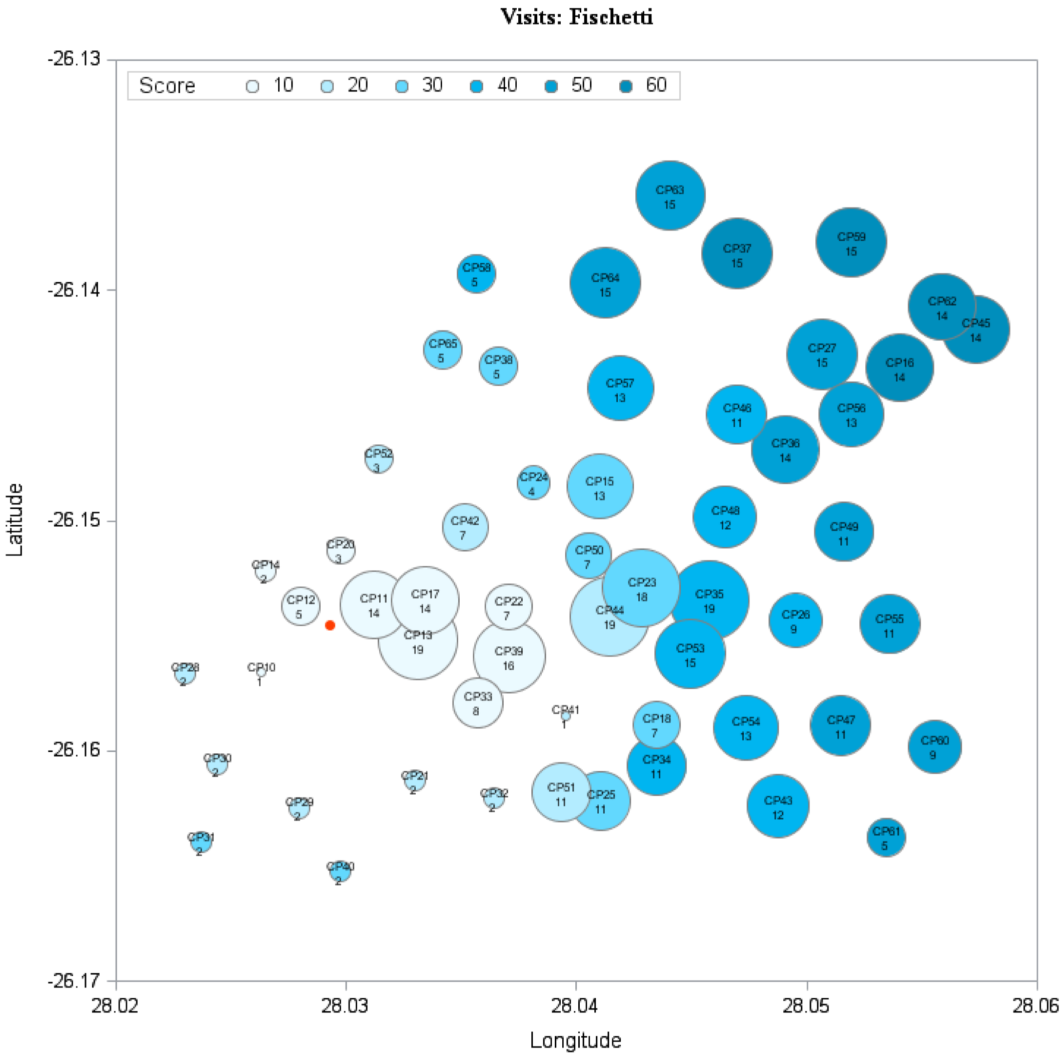


Figure L.26: Visit frequencies resulting from Fischetti